

Об одном подходе к организации домашней распределенной вычислительной системы

С.А. Балабаев, С.А. Лупин, Д.А. Федяшин

Аннотация— В работе рассмотрен подход к организации домашней распределенной вычислительной системы, состоящей из персонального компьютера и интегрированных с ним смартфона, микрокомпьютера Raspberry Pi и телевизионной приставки Smart TV. Дан обзор существующих программных решений – кластеров из смартфонов и микрокомпьютеров Raspberry Pi. Отмечается, что использование маломощных устройств в качестве узлов распределенной системы возможно, однако сопровождается сложностями в разработке и настройке программного обеспечения. По результатам анализа был определен функционал и разработано ПО для интеграции мобильных устройств с ПК. Оно позволяет загружать на узлы расчетную задачу, запускать ее, накапливать и выводить полученные результаты вычислений на экран. Взаимодействие между устройствами происходит по локальной сети. ПО для мобильных устройств разработано на языке Java. В работе предлагается способ запуска скриптов, написанных на языке Python на смартфонах с помощью приложения Termux. Используемый алгоритм балансировки нагрузки узлов позволяет объединять в единую среду устройства, существенно отличающиеся по производительности. Представлены результаты решения задачи федеративного обучения нейронной сети в распределенной среде, организованной с помощью разработанного программного обеспечения. Они подтверждают функциональность разработанного ПО и возможность использования организованной с её помощью распределенной системы с разнородными узлами для решения оптимизационных задач.

Ключевые слова— распределенные вычисления, грид системы, смартфон, ОС Android, Raspberry Pi, Smart TV.

I. ВВЕДЕНИЕ

В современном мире существуют разнообразные задачи, такие как поиск новых звезд, открытие структур белка, расчет прогноза погоды. Для их эффективного решения необходимо использовать высокопроизводительные системы - суперкомпьютеры, кластеры, гриды. Однако, начинающему исследователю сложно получить доступ к подобным системам, поэтому для проведения экспериментов он вынужден использовать домашний персональный компьютер. Чтобы повысить эффективность вычислений возможно объединить его с устройствами, которые уже есть в доме – смартфонами, телевизионными приставками или элементами умного дома. Организованная таким образом распределенная система относится к классу Home GRID (HGRID) – домашний грид. Её особенностью является

высокая гетерогенность узлов, поэтому для повышения эффективности вычислений необходимо балансировать нагрузку в зависимости от мощности каждого из входящих в неё устройств.

II. СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

Все решения по объединению разнородных устройств в гриды возможно объединить в несколько классов. Первый класс предусматривает объединение одинаковых по назначению устройств с различными характеристиками. К нему можно отнести гриды из персональных компьютеров, гриды из смартфонов, гриды из микрокомпьютеров Raspberry Pi.

Применение гридов из смартфонов для решения сложных задач является одним из популярных направлений исследований [1]. Современные смартфоны обладают высокими характеристиками и способны выполнять высокопроизводительные вычисления [2]. Основными особенностями использования смартфонов в качестве узлов распределенной системы являются: необходимость их постоянного подключения к электропитанию и дополнительного охлаждения, ограниченность возможностей запуска сторонних приложений, гетерогенность системы [3].

Следует учитывать, что при проведении интенсивных вычислений происходит высокое потребление энергии и быстрый разряд аккумулятора и, как следствие, замедление работы. Аналогично при продолжительных расчетах температура процессора повышается, что приводит к автоматическому снижению тактовой частоты процессора. Для разработки приложений под устройства с операционной системой Android необходимо использовать языки программирования Java или Kotlin. Использование C/C++ для написания участков кода, которые описывают алгоритмы работы, возможно только вместе с указанными выше языками. Из-за сильных различий в характеристиках устройств применение стандартных методов распараллеливания программ приведет к низкой их эффективности. Поэтому важно применять алгоритмы балансировки, учитывающие характеристики узлов [4].

Распределенная система разрабатывается по принципу клиент-серверной архитектуры. Задачи передаются с основного устройства на узлы и выполняются на них. Функционал приложений не имеет формальных ограничений, поэтому в среде HGRID мы можем решать

самые разные задачи, например обучение нейронной сети. Примером проекта, реализующего первый подход, является Talos – приложение, предназначенное для обучения нейронных сетей [5]. В работах [6,7,8,9] предлагается использовать смартфоны в качестве узлов для обучения нейронных сетей, однако предложенная архитектура приложения позволяет решать только задачу обучения. В качестве примеров проекта с универсальным подходом являются BOINC, IBIS и CWC.

Самым популярным из всех существующих решений применения мобильных устройств для высоконагруженных вычислений является проект BOINC. Это программный комплекс для организации добровольных распределённых вычислений. С его помощью можно рассчитывать сложные задачи, используя ресурсы персональных компьютеров добровольцев, подключившихся к проекту. Желающие принять участие в проекте должны установить на свой ПК специальное ПО, позволяющее подключиться к системе. Далее часть вычислительной задачи будет отправлена пользователю и рассчитана в свободное от нагрузки время [10, 11, 12, 13].

Первоначально BOINC разрабатывался в США для проекта добровольных вычислений SETI@home, производящего поиск внеземного разума, но впоследствии разработчики сделали платформу доступной и для сторонних проектов.

Работа системы происходит следующим образом:

1. Клиент и сервер запускают приложение.
2. Клиент выбирает проект, в котором он хочет принять участие.
3. С сервера проекта клиенту передаются инструкции и исполняемый файл.
4. На клиентском узле автоматически происходит запуск исполняемого файла в соответствии с полученными инструкциями.
5. После решения задачи клиент отправляет решение на сервер.

Особенности решения:

- Простота включения в проект ресурсов

добровольцев.

- Простота интерфейса.
- Высокий порог вхождения для разработки собственных приложений.
- Иностранная разработка.

Проект BOINC очень удобен для клиента – установка и настройка системы не предполагает высоких затрат. Однако, использование системы в качестве собственной платформы для вычислений может вызвать затруднение у исследователей, не имеющих высокого навыка программирования. Обращает на себя внимание и тот факт, что, хотя BOINC и является проектом с открытым исходным кодом, его разработчики рассматривают возможность бойкотировать российские проекты [14].

Нидерландские ученые разработали систему IBIS, позволяющую объединять различные устройства для распределённых вычислений. Проект написан на языке Java и предоставляет возможность работы с Android устройствами. Авторы предлагают эффективный способ коммуникации устройств между собой с помощью разработанной библиотеки и поддержку MPI. Однако в 2014 году проект был заморожен и доступа к нему нет [15,16].

Американскими исследователями был разработан проект CWC - Computing while charging. Предложено внедрить в крупных компаниях проведение высокопроизводительных вычислений на гриде из смартфонов вместо серверов и облачных вычислений. Каждый сотрудник компании должен предоставлять мощности своего устройства во время его бездействия – ночью при зарядке. Эксперименты показали работоспособность проекта. Показано, что четыре смартфона, объединенных в сеть сопоставимы по мощностям с персональным компьютером [17, 18].

Для решения ресурсоемких задач можно использовать и грид из Raspberry Pi – популярной платформы, на основе которой разрабатываются устройства для умного дома, роботы, медиацентры, DIY-проекты [19]. Существует несколько версий микрокомпьютеров, обладающих разными характеристиками (табл. 1).

Таблица 1: Сравнение качества моделей

Параметр	Raspberry Pi				
	Model 1B	Model 2B	Model 3B	Model 4B	Model 5B
Число ядер	1	4	4	4	4
Тактовая частота (GHz)	0.7	0.9	1.2-1.4	1.5	2.4
RAM (Gb)	0.5	1	1	1-4	2-8

В работе [20] рассказывается о применении кластера из Raspberry Pi для машинного обучения. Разработанный исследователями кластер состоит из 6 устройств Raspberry Pi Model 4B с 4 ГБ оперативной памяти на каждом узле. Устройства были подключены к коммутатору Gigabit Ethernet. Кластер использует распределенную файловую систему Hadoop (3.2.0) и Apache Spark (2.4.3) для реализации алгоритмов распределенного машинного обучения (MLlib Spark). Показано, что производительность кластера достаточна для машинного обучения.

Одной из сфер применения грид из Raspberry Pi является образование, когда студенты обучаются практической работе с распределенными системами [21, 22]. В работах подчеркивается простота проектов и их относительная низкая цена.

Проведенный анализ существующих подходов, позволяет констатировать, что на рынке нет универсального решения, и задачи создания ПО для организации HGRID является актуальной, особенно в условиях санкционных ограничений.

III. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

Программное обеспечение для организации HGRID разрабатывается и учетом того, что будут выполняться следующие условия [23]:

1. Устройства подключаются к локальной сети с использованием точек WiFi;
2. Все устройства подключены к сети электропитания;
3. Устройства на время вычисления не могут быть отключены или вынесены вне зоны WiFi.

Функционал интегрирующего ПО должен обеспечивать возможность объединения устройств с существенно разнородными характеристиками.

В качестве тестовой была выбрана платформа, состоящая из персонального компьютера и трех устройств – смартфона, микрокомпьютера Raspberry Pi и телевизионной приставки Smart-TV. (Рис. 1)

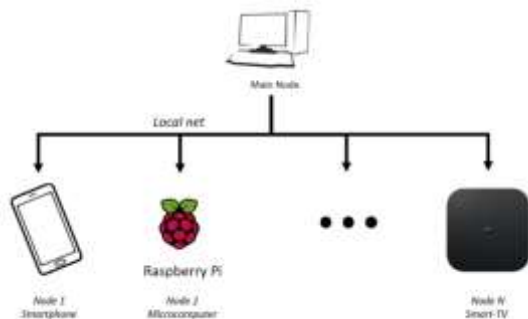


Рис. 1: Структура HGRID

Разработанная система работает по следующему алгоритму:

1. Подключение всех устройств.
2. Проверка наличия узла в списке ранее использовавшихся.
3. В случае отсутствия – проведение тестирования и вычисление коэффициента его относительной мощности.
4. Разбиение задачи на части в соответствии с коэффициентами относительной мощности и передача её на узлы.
5. Прием от узлов метрик, характеризующих режим работы. В случае превышения предельных параметров - отключение узла.
6. Вывод результата вычислений на экран основного устройства после их окончания.

A. Архитектура ПО

После отправления задачи с сервера она поступает по локальной сети в приложение клиента. Если в качестве клиента выступает смартфон на платформе Android или телевизионная приставка Smart-TV, то в качестве основного процесса, принимающего сообщения от сервера является приложение Android, написанное на языке java. Основными задачами приложения являются определение типа задачи, её запуск и мониторинг характеристик устройства. Если задача была оформлена в виде ark-приложения, то она запускается с помощью библиотечных вызовов самого приложения. Если задача представляет собой скрипт, разработанный, например, на языке python, то данные передаются в приложение, разработанное под эмулятор командной строки Termux. Внутри него происходит запуск программы и отправка результата обратно на клиентское приложение, которое в свою очередь возвращает результат вычисления на сервер.

Приложение Termux свободно лицензируется и может устанавливаться на большинство современных мобильных устройств [24]. Его удобно использовать для запуска простейших приложений на C, C++, Python и других языках программирования. Недостаток связан с отсутствием многих распространенных библиотек. Для решения проблемы возможно внутри приложения установить образ дистрибутива ubuntu, на который загрузить необходимое программное обеспечение. В таблице 2 представлено сравнение наличия основных библиотек, необходимых для машинного обучения на Termux и в виртуальной машине ubuntu. Недостатком использования ubuntu является сильное замедление работы программы, и невозможность использования всех ядер процессора.

Таблица 2: Доступность библиотек машинного обучения

Библиотека	Termux	Ubuntu on Termux
numpy	python-numpy	+
tensorflow	-	+
keras	-	+
pytorch	-	+

В том случае, когда в качестве клиента выступает микрокомпьютер, переданная задача может быть скомпилирована, обработана и запущена, как и на любом обычном персональном компьютере. При этом есть возможность использование контейнеризации для запуска приложений.

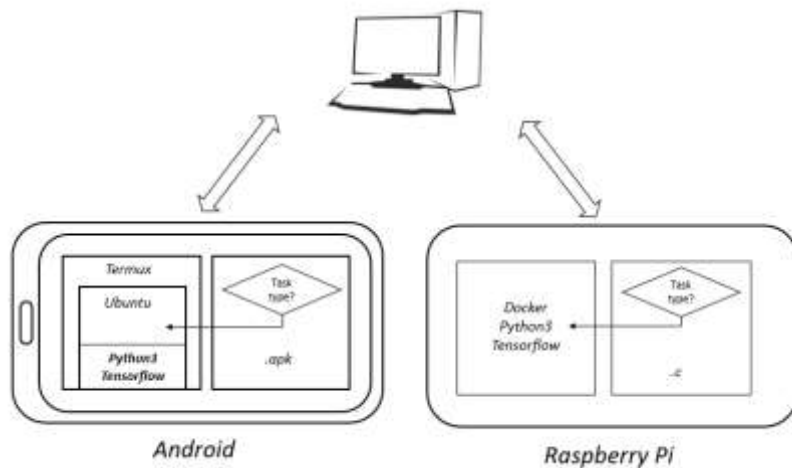


Рис. 2: Архитектура системы HGRID.

IV. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

Для оценки вычислительных возможностей HGRID были проведены эксперименты по федеративному обучению нейронной сети для распознавания рекламных сообщений при рассылке по электронной почте. В качестве исходного датасета были выбраны 22032 корректных сообщения и 9000 писем со спамом [25].

Характеристики узлов HGRID, выбранных для проведения экспериментов, представлены в таблице 3.

Таблица 3: Характеристики узлов HGRID

Устройство	Характеристики
PC-1	Intel core i5-11400H @2.70 GHz 6 core 12 threads 24 Gb RAM
PC-2	Intel core i5-5200U @2.2 GHz 2 core 6 Gb RAM

RPI	Raspberry Pi 4 Model B Cortex-A72 @1.8 GHz 4 core 6 Gb RAM
Smartphone	Xiaomi Redmi ARM 2× Cortex-A78 @2.6 GHz, ARM 6× Cortex-A55 @2.0 GHz 8 cores, 8 Gb RAM
Smart-TV	Rgeed Cortex-A53 @1.51 GHz 4 cores, 4 Gb RAM
Smartphone	Intel core i5-11400H @2.70 GHz 6 core 12 threads 24 Gb RAM

В первом эксперименте измерялось время обучения сети отдельно на каждом из узлов в однопоточном и многопоточном режимах при использовании алгоритма балансировки нагрузки на узлах. Разработанное приложение не использовало сторонние библиотеки, поэтому его запуск был возможен без использования эмулятора *ubuntu*. Результаты эксперимента приведены в таблице 4.

Таблица 4: Результаты первого эксперимента

Время вычисления (сек)	PC-1	PC-2	Raspberry Pi	Smartphone		Smart-TV	
				UBUNTU ON TERMUX	TERMUX	UBUNTU ON TERMUX	TERMUX
Максимум потоков	4.2	48.1	46.36	96.6	19.6	-	109.8
1 поток	23.3	71.8	181.2	98.14	55.8	-	433.2

Можно заметить, что время решения задачи на смартфоне и Raspberry Pi ниже, чем на персональном компьютере PC-2. Это объясняется наличием у них большего числа вычислительных ядер, чем на компьютере и его низкой производительностью. Из-за того, что процессор Smart-TV был 32 битной версии, не удалось полноценно запустить на нём *ubuntu*. Время выполнения задачи в однопоточном и многопоточном режимах на *ubuntu* на смартфоне совпало, что подтверждает теорию о невозможности использования нескольких ядер.

Во втором эксперименте аналогичная задача была запущена на всех узлах HGRID, в состав которой были по очереди включены PC-1 и PC-2. При использовании ресурсов HGRID время обучения удается уменьшить в 1.34 раза по сравнению с обучением только на PC-1. В

случае с PC-2 использованием распределенной системы ускоряет процесс обучения в 4,67 раз (Таблица 5).

Таблица 5: Результаты второго эксперимента

HGRID	Время обучения сети (сек)		
	PC	HGRID	Ускорение
PC-1(балансировка)	4.2	3,13	1,34
PC-2(балансировка)	48.1	10,3	4,67
PC-2	48.1	27,6	1,73

При запуске вычислительной задачи на всех устройствах, для повышения эффективности был применен метод балансировки, учитывающий коэффициент относительной мощности узла. На рисунках представлены круговые диаграммы, отражающие вклад каждого из узлов в решение задачи. Можно заметить, что мощный персональный компьютер

PC-1 выполнил $\frac{3}{4}$ всей работы, когда PC-2 в составе HGRID только $\frac{1}{4}$ (Рис. 3-4).

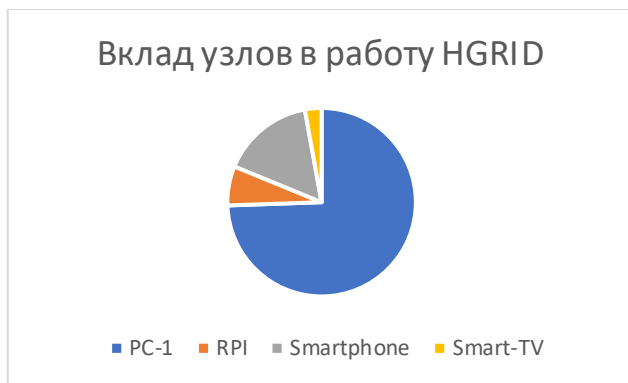


Рис. 3: Вклад узлов в работу HGRID (первый эксперимент)

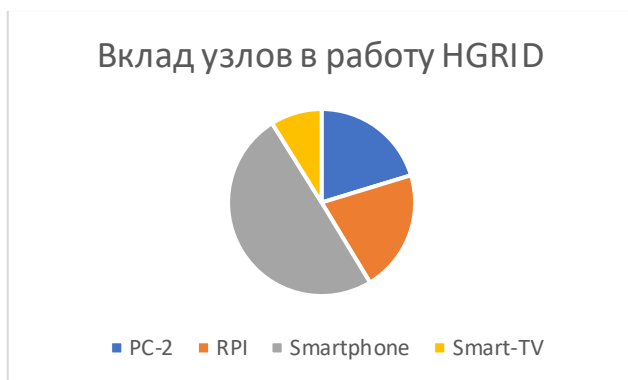


Рис. 4: Вклад узлов в работу HGRID (второй эксперимент)

Третья строка в Табл. 6 позволяет оценить эффективность балансировки нагрузки узлов. Без неё время обучения сети увеличилось в 2,67 раз.

V. ВЫВОДЫ

Предлагаемый подход к организации домашней распределенной вычислительной системы в ходе экспериментов подтвердил свою функциональность на практике. Даже небольшой HGRID позволяет уменьшить время обучения нейронной сети, особенно в случае применения балансировки нагрузки узлов. Было показано, что использование приложения Termux и установленного на него образа ubuntu делает предложенное решение универсальным. Следует отметить, что хотя домашняя сеть и не может конкурировать с кластерами из рабочих станций, её можно использовать не только для отладки распределенных приложений, но и для расширения функционала IoT систем.

БИБЛИОГРАФИЯ

- [1] Pramanik P. K. D., Pal S., Choudhury P. Mobile crowd computing potential, architecture, requirements, challenges, and applications // *The Journal of Supercomputing*. – 2024. – Т. 80. – №. 2. – С. 2223-2318. URL: <https://doi.org/10.1007/s11227-023-05545-0>
- [2] С. А. Балабаев, «Оценка вычислительных возможностей мобильных платформ», 28-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов «Микроэлектроника и информатика - 2021», 2021.
- [3] Балабаев С. А., Лупин С. А., Тайк А. М. Система мониторинга для балансировки нагрузки узлов распределенной вычислительной системы на основе смартфонов // *International Journal of Open Information Technologies*. – 2024. – Т. 12. – №. 10. – С. 78-85.
- [4] Кхаинг М., Лупин С. А., Тху А. Оценка эффективности методов балансировки нагрузки в распределенных вычислительных системах // *International Journal of Open Information Technologies*. – 2021. – Т. 9. – №. 11. – С. 30-36.
- [5] Takawale H. C., Thakur A. Talos app: on-device machine learning using tensorflow to detect android malware // *2018 fifth international conference on Internet of Things: systems, management and security*. – IEEE, 2018. – С. 250-255. URL: <http://dx.doi.org/10.1109/IoTSMS.2018.8554572>
- [6] Salem H. Distributed computing system on a smartphones-based network // *Software Technology: Methods and Tools: 51st International Conference, TOOLS 2019, Innopolis, Russia, October 15–17, 2019, Proceedings 51*. – Springer International Publishing, 2019. URL: http://dx.doi.org/10.1007/978-3-030-29852-4_26
- [7] Kaushik P., Yadav P. K. A novel approach for detecting malware in android applications using deep learning // *2018 Eleventh International Conference on Contemporary Computing (IC3)*. – IEEE, 2018. – С. 1-4. <https://doi.org/10.1109/IC3.2018.8530668>
- [8] Fang W. et al. Comprehensive android malware detection based on federated learning architecture // *IEEE Transactions on Information Forensics and Security*. – 2023. – Т. 18. – С. 3977-3990. <https://doi.org/10.1109/TIFS.2023.3287395>
- [9] Tang J. et al. PE-FedAvg: A Privacy-Enhanced Federated Learning for Distributed Android Malware Detection // *2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom)*. – IEEE, 2023. – С. 474-481 <https://doi.org/10.1109/ISPA-BDCLOUD-SocialCom-SustainCom59178.2023.00094>
- [10] Kurochkin I. et al. Using Mobile Devices in a Voluntary Distributed Computing Project to Solve Combinatorial Problems // *Supercomputing: 7th Russian Supercomputing Days, RuSCDays 2021, Moscow, Russia, September 27–28, 2021, Revised Selected Papers 7*. – Springer International Publishing, 2021. – С. 525-537. http://dx.doi.org/10.1007/978-3-030-92864-3_40
- [11] Долгов А. А. Разворачивание Грид-системы из Мобильных устройств на платформе BOINC // *Облачные и распределенные вычислительные системы в электронном управлении ОРВСЭУ-2022 в рамках национального суперкомпьютерного форума (НСКФ-2022)*, 2022 с. 24-29
- [12] Курочкин И. И., Прун А. И. Грид-система из персональных устройств на платформе BOINC для решения задач глубокого обучения // *Опτικο-электронные приборы и устройства в системах распознавания образов и обработки изображений: сборник материалов XVII Международной научно-технической конференции, Курск, 12–15 сентября 2023 года*. Курск: Юго-Западный государственный университет, 2023. С. 252-254.
- [13] Курочкин И. И. Децентрализованное глубокое обучение на грид-системе из персональных компьютеров // *Материалы XXIII Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2023)*, с. Дивноморское, Краснодарский край, 04–10 сентября 2023 года. Москва: Московский авиационный институт (национальный исследовательский университет), 2023. С. 111-113.
- [14] Официальный сайт проекта BOINC [Электронный ресурс] / URL: <https://boinc.berkeley.edu/russia.php> (Дата обращения: 30.11.24)
- [15] Gurusamy V., Nandhini K. International journal of engineering sciences & research technology IBIS: The new era for distributed computing. <http://dx.doi.org/2018.10.5281/zenodo.1135392>
- [16] Palmer N. et al. Ibis for mobility: solving challenges of mobile computing using grid techniques // *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*. – 2009. – С. 1-6. <http://dx.doi.org/10.1145/1514411.1514426>
- [17] Kumar T. U., Senthilkumar R. CWC* — Secured distributed computing using Android devices // *2016 International Conference on Recent Trends in Information Technology (ICRTIT)*. – IEEE, 2016. – С. 1-7 <https://doi.org/10.1109/ICRTIT.2016.7569590>
- [18] Arslan M. Y. et al. Computing while charging: Building a distributed computing infrastructure using smartphones // *Proceedings of the 8th International conference on Emerging networking experiments and technologies*. – 2012. – С. 193-204. <https://doi.org/10.1145/2413176.2413199>
- [19] Балабаев С. А., Лупин С. А., Шакиров Р. Н. Вычислительный кластер на основе смартфонов Android и микрокомпьютеров

- Raspberry Pi //International Journal of Open Information Technologies. – 2022. – Т. 10. – №. 7. – С. 86-93.
- [20] Komninos A. et al. Performance of raspberry pi microclusters for edge machine learning in tourism //Network (Mbps). – 2019. – Т. 100. – №. 100. – С. 100.
- [21] Xu Z. Teaching heterogeneous and parallel computing with google colab and raspberry pi clusters //Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing Network, Storage, and Analysis. – 2023. – С. 308-313 <https://doi.org/10.1145/3624062.3624095>
- [22] Govindaraj, Parallel Programming in Raspberry Pi Cluster, Ithaca, 2016
- [23] Балабаев С.А., Лупин С.А. Оценка функциональности кластера из персонального компьютера и мобильных устройств / IV научно-практическая конференция с международным участием «Актуальные проблемы информатизации в цифровой экономике и научных исследованиях – 2023», Зеленоград. 2023.
- [24] Балабаев С.А., Гуреев А.В. Сравнение методов разработки гетерогенного кластера из мобильных устройств на платформе Android. // 29-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов «Микроэлектроника и информатика - 2022» Зеленоград. 2022.
- [25] Балабаев С.А., Балабаев А.А., Применение CoMD систем для обучения нейронных сетей // 31-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов «Микроэлектроника и информатика - 2024» Зеленоград. 2024.

About one approach to organizing a home distributed computing system

Sergey Balabaev, Sergey Lupin, Dmitry Fedyashin

Abstract - The paper considers an approach to organizing a home distributed computing system consisting of a personal computer and an integrated smartphone, a Raspberry Pi microcomputer, and a Smart TV set-top box. An overview of existing software solutions is given - clusters of smartphones and Raspberry Pi microcomputers. It is noted that the use of low-power devices as nodes of a distributed system is possible, but is accompanied by difficulties in developing and configuring software. Based on the analysis, the functionality was determined and software for integrating mobile devices with a PC was developed. It allows you to upload a calculation task to the nodes, run it, accumulate and display the obtained calculation results on the screen. Interaction between devices occurs over a local network. The software for mobile devices is developed in Java. The paper proposes a method for running scripts written in Python on smartphones using the Termux application. The used node load balancing algorithm allows you to combine devices with significantly different performance into a single environment. The article presents the results of solving the problem of federated training of a neural network in a distributed environment organized using the developed software. They confirm the functionality of the developed software and the possibility of using a distributed system organized with its help with heterogeneous nodes to solve optimization problems.

Keywords - distributed computing, smartphone, Android OS, grid systems, Raspberry Pi, Smart TV.

REFERENCES

- [1] Pramanik P. K. D., Pal S., Choudhury P. Mobile crowd computing potential, architecture, requirements, challenges, and applications //The Journal of Supercomputing. - 2024. - Vol. 80. - No. 2. - P. 2223-2318. URL: <https://doi.org/10.1007/s11227-023-05545-0>
- [2] S. A. Balabaev, "Evaluation of computing capabilities of mobile platforms," 28th All-Russian Interuniversity Scientific and Technical Conference of Students and Postgraduates "Microelectronics and Computer Science - 2021", 2021.
- [3] Balabaev S. A., Lupin S. A., Taik A. M. Monitoring system for load balancing nodes of a distributed computing system based on smartphones //International Journal of Open Information Technologies. - 2024. - Vol. 12. - No. 10. - P. 78-85.
- [4] Khaing M., Lupin S. A., Thu A. Evaluating the effectiveness of load balancing methods in distributed computing systems // International Journal of Open Information Technologies. - 2021. - Vol. 9. - No. 11. - P. 30-36.
- [5] Takawale H. C., Thakur A. Talos app: on-device machine learning using tensorflow to detect android malware // 2018 fifth international conference on Internet of Things: systems, management and security. - IEEE, 2018. - P. 250-255. URL: <http://dx.doi.org/10.1109/IoTSMS.2018.8554572>
- [6] Salem H. Distributed computing system on a smartphones-based network //Software Technology: Methods and Tools: 51st International Conference, TOOLS 2019, Innopolis, Russia, October 15–17, 2019, Proceedings 51. - Springer International Publishing 2019 URL: http://dx.doi.org/10.1007/978-3-030-29852-4_26
- [7] Kaushik P., Yadav P. K. A novel approach for detecting malware in android applications using deep learning //2018 Eleventh International Conference on Contemporary Computing (IC3). - IEEE, 2018. - pp. 1-4. <https://doi.org/10.1109/IC3.2018.8530668>
- [8] Fang W. et al. Comprehensive android malware detection based on federated learning architecture //IEEE Transactions on Information Forensics and Security. - 2023. - T. 18. - P. 3977-3990. <https://doi.org/10.1109/TIFS.2023.3287395>
- [9] Tang J. et al. PE-FedAvg: A Privacy-Enhanced Federated Learning for Distributed Android Malware Detection //2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom). - IEEE, 2023. - pp. 474-481 <https://doi.org/10.1109/ISPA-BDCLOUD-SocialCom-SustainCom59178.2023.00094>
- [10] Kurochkin I. et al. Using Mobile Devices in a Voluntary Distributed Computing Project to Solve Combinatorial Problems //Supercomputing 7th Russian Supercomputing Days, RuSCDays 2021, Moscow, Russia, September 27–28, 2021, Revised Selected Papers 7. - Springer International Publishing, 2021. - pp. 525-537. http://dx.doi.org/10.1007/978-3-030-92864-3_40
- [11] Dolgov A. A. Deployment of a Grid System from Mobile Devices on the BOINC Platform // Cloud and Distributed Computing Systems in Electronic Management of ORVSEU-2022 within the Framework of the National Supercomputer Forum (NSCF-2022), 2022 pp. 24-29
- [12] Kurochkin I. I., Prun A. I. Grid System from Personal Devices on the BOINC Platform for Solving Deep Learning Problems // Optical-electronic Devices and Devices in Pattern Recognition and Image Processing Systems: Collection of Materials of the XVII International Scientific and Technical Conference, Kursk, September 12-15, 2023. Kursk: South-West State University, 2023. pp. 252-254.
- [13] Kurochkin I. I. Decentralized deep learning on a grid system of personal computers // Proceedings of the XXIII International Conference on Computational Mechanics and Modern Applied Software Systems (VMSPPS'2023), p. Divnomorskoye, Krasnodar Krai, September 4-10, 2023. Moscow: Moscow Aviation Institute (National Research University), 2023. Pp. 111-113.
- [14] Official website of the BOINC project [Electronic resource] / URL: <https://boinc.berkeley.edu/russia.php> (Accessed: 11/30/24)
- [15] Gurusamy V., Nandhini K. International journal of engineering sciences & research technology IBIS: The new era for distributed computing <http://dx.doi.org/2018.10.5281/zenodo.1135392>
- [16] Palmer N. et al. Ibis for mobility: solving challenges of mobile computing using grid techniques //Proceedings of the 10th workshop on Mobile Computing Systems and Applications. - 2009. - P. 1-6. <http://dx.doi.org/10.1145/1514411.1514426>
- [17] Kumar T. U., Senthilkumar R. CWC* - Secured distributed computing using Android devices //2016 International Conference on Recent Trends in Information Technology (ICRTIT). - IEEE, 2016. - pp. 1-7 <https://doi.org/10.1109/ICRTIT.2016.7569590>
- [18] Arslan M. Y. et al. Computing while charging: Building a distributed computing infrastructure using smartphones //Proceedings of the 8th International conference on Emerging networking experiments and technologies. - 2012. - P. 193-204. <https://doi.org/10.1145/2413176.2413199>
- [19] Balabaev S. A., Lupin S. A., Sha Kirov R. N. Computing cluster based on Android smartphones and Raspberry Pi microcomputers // International Journal of Open Information Technologies. - 2022. - Vol. 10. - No. 7. - P. 86-93.
- [20] Komninos A. et al. Performance of raspberry pi microclusters for edge machine learning in tourism // Network (Mbps). - 2019. - Vol. 100. - No. 100. - P. 100.
- [21] Xu Z. Teaching heterogeneous and parallel computing with google colab and raspberry pi clusters // Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis. - 2023. - P. 308-313 <https://doi.org/10.1145/3624062.3624095>
- [22] Govindaraj, Parallel Programming in Raspberry Pi Cluster, Ithaca, 2016
- [23] Balabaev S. A., Lupin S. A. Evaluation of the functionality of a cluster of a personal computer and mobile devices / IV scientific and practical conference with international participation "Actual problems of informatization in the digital economy and scientific research - 2023", Zelenograd. 2023.

- [24] Balabaev S.A., Gureev A.V. Comparison of methods for developing a heterogeneous cluster of mobile devices on the Android platform. // 29th All-Russian Interuniversity Scientific and Technical Conference of Students and Postgraduates "Microelectronics and Computer Science - 2022" Zelenograd. 2022.
- [25] Balabaev S.A., Balabaev A.A., Application of CoMD systems for training neural networks // 31st All-Russian Interuniversity Scientific and Technical Conference of Students and Postgraduates "Microelectronics and Computer Science - 2024" Zelenograd. 2024.