

Долгов¹ А.А., Курочкин^{1,2} И.И.

¹Национальный исследовательский технологический университет МИСИС
119049, Москва, Ленинский пр-кт, д. 4, стр. 1

²Институт проблем передачи информации им. А.А. Харкевича РАН
127051, г. Москва, Большой Каретный переулок, д.19 стр. 1

E-mail: kurochkin@iitp.ru

Использование мобильных устройств в грид-системах из персональных компьютеров

Распределенные вычисления для решения научных вычислительных задач набрали популярность в последние десятилетия. Распределенные вычислительные системы или грид-системы позволяют агрегировать географически разнесенные и разнородные ресурсы от разных организаций для решения комплексных вычислительных задач. В связи с популярностью использования мобильных устройств в последние годы, а также стремительным ростом вычислительных возможностей этих устройств, смартфоны и планшеты способны проводить ресурсоемкие операции, что позволяет рассматривать их как поставщиков вычислительных ресурсов. В течение последнего десятилетия во многих исследовательских проектах была решена проблема включения мобильных устройств в сеть. Появилось различное программное обеспечение для организации грид-систем, например BOINC – программная платформа с открытым исходным кодом. Платформа BOINC позволяет разворачивать проекты распределенных вычислений, где в качестве вычислительных узлов могут использоваться, как персональные компьютеры, ноутбуки, так и мобильные устройства на базе ОС Android.

Был развернут тестовый проект распределенных вычислений parlea.ru/boinctest, в котором в качестве вычислительных узлов могут выступать как мобильные устройства, так и персональные компьютеры, и ноутбуки. Проект охватывает архитектуры процессоров: x86, aarch64, arm, x86-64. Вычислительное приложение, решающее одну из многочисленных задач по поиску ортогональных диагональных латинских квадратов [1], доступно для нескольких операционных систем, а именно: Windows, Linux, Android. На данном проекте было проведено несколько вычислительных экспериментов с различными серверными параметрами на грид-системе из персональных устройств разных типов. Произведена настройка параметров серверной части проекта распределенных вычислений, которая позволила уменьшить время проведения вычислительных экспериментов, увеличить процент загрузки устройств, уменьшить процент просроченных заданий [2].

На текущий момент на проекте запущен вычислительный эксперимент, длительностью в несколько недель, в котором активно участвует около 300 различных устройств. Динамику подключения узлов к проекту можно увидеть на рис. 1.

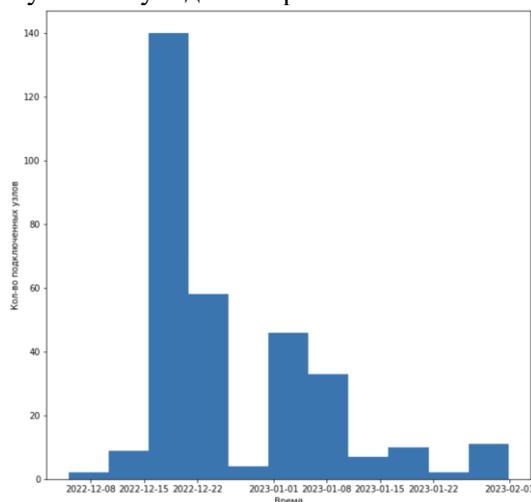


Рис.1. Динамика подключения узлов к проекту.

В текущем эксперименте необходимо просчитать более миллиона рабочих единиц разной сложности (от нескольких секунд до нескольких часов). Исходя из этого, необходимо увеличить количество получаемых и запаасаемых задач на узлах. Для этого были убраны жесткие ограничения у параметров `max_wus_to_send` и `max_wus_in_progress` повысив их значение до 200 и 100 соответственно. Также в рамках текущего эксперимента была разблокирована возможность получения одним хостом одного юзера более одного результата конкретной рабочей единицы (параметры `one_result_per_user_per_wu` и `one_result_per_host_per_wu`) для того, чтобы повысить темп возвращаемых результатов. В публичном проекте этого делать не стоит, поскольку возможны намеренно испорченные результаты со стороны пользователей. Было установлено, что часть возвращаемых результатов (около 5%), были возвращены на сервер с ошибкой, связанной с неправильным ограничением максимальной вычислительной мощности для рабочей единицы. Параметр `rsc_flops_bound` был повышен на несколько порядков до $8.25 \cdot 10^{25}$ для того, чтобы свести к минимуму влияние таких результатов на общий эксперимент. Минимальный размер кворума, а также параметр репликации: `min_quorum`, `target_nresults` равняются 2 соответственно, чтобыкратно не увеличивать множество рабочих единиц, но при этом сохранить возможность валидирования результатов. Параметр крайнего срока выполнения(`deadline`) `delay_bound` остался неизменным и численно равен 5 дням.

Чтобы избежать потери большого массива данных, полученных в результате вычислительного эксперимента, на проекте предусмотрен механизм двухэтапного бекапа, дважды в день архивированные результаты сохраняются на внешний жесткий диск и дважды в сутки результаты выгружаются с сервера проекта.

В процессе эксперимента дополнительно были использованы возможности пакета `boinc2docker`, который использует `Docker` контейнеры в качестве оберток для вычислительных приложений. Данное решение было интегрировано в проект и запущено асинхронно второе вычислительное приложение с несколькими сотнями простых задач. Все множество задач было успешно посчитано и не возникло конфликтов с первым вычислительным приложением. Этот механизм значительно упрощает работу, необходимую для разработки и развертывания приложений для BOINC. Он работает, комбинируя возможности `boot2docker` и `vboxwrapper`. Главные плюсы данного решения:

- позволяет избежать использования BOINC API в вычислительном приложении;
- контрольная точка автоматически предоставляется `vboxwrapper`;
- вычислительное приложение автоматически работает в Windows, Mac OS и Linux;
- можно удобно создавать свои приложения с помощью `Dockerfiles`.

На основе полученного результата дополнительно планируется реализация вычислительного приложения для решения задачи распределенного обучения нейронной сети.

Полученный опыт, а также подобранные параметры в процессе эксперимента, можно применять в организации комплексных вычислительных экспериментов на грид-системах на базе BOINC, с приоритетом не только на персональные компьютеры, но и на мобильные устройства.

Литература

1. Vatutin E., Belyshev A., Kochemazov S., Zaikin O., Nikitina N. Enumeration of isotopy classes of diagonal Latin squares of small order using volunteer computing // *Communications in Computer and Information Science*. Vol. 965, Springer, 2018. pp. 578–586.
2. Kurochkin, I., Dolgov, A., Manzyuk, M., Vatutin, E. Using mobile devices in a voluntary distributed computing project to solve combinatorial problems // In: Voevodin, V., Sobolev, S. (eds) *Supercomputing. RuSCDays 2021. Communications in Computer and Information Science*, Vol 1510, Springer, Cham, 2021. pp. 525-537.