

## **О некоторых особенностях использования вычислительных кластеров в проектах добровольных распределенных вычислений**

Заикин О.С.<sup>1</sup>, Манзюк М.О.<sup>2</sup>, Булавинцев В.Г.<sup>1</sup>, Посыпкин М.А.<sup>3</sup>

Институт динамики систем и теории управления им. В.М.

Матросова СО РАН, Иркутск<sup>1</sup>

Интернет-портал BOINC.ru<sup>2</sup>

Институт проблем передачи информации им. А.А. Харкевича РАН,  
Москва<sup>3</sup>

За последние 20 добровольные распределенные вычисления [1, 2] были успешно использованы для решения разнообразных научных задач, допускающих разбиение на независимые подзадачи. В частности, в российском проекте SAT@home [3], запущенном совместно ИДСТУ СО РАН и ИППИ РАН в 2011 году, были успешно решены задачи анализа стойкости некоторых поточных шифров, а также задачи поиска новых систем латинских квадратов.

Для увеличения производительности проекта SAT@home в 2013 году была разработана первая версия программного средства CluBORun (Cluster for BOINC Run). Данное средство позволяет запускать свободных узлах вычислительных кластеров задания любых проектов добровольных распределенных вычислений, основанных на платформе BOINC. Основными принципами работы CluBORun перечислены ниже.

1. Для работы CluBORun используются только стандартные права пользователя кластера.
2. Узлы кластера воспринимаются проектом добровольных вычислений как обычные ПК.
3. Для запуска BOINC на узле кластера используется стандартный BOINC-менеджер, который задействует все ресурсы узла.
4. Для запуска BOINC кластере используется специальная MPI-программа, которая на каждом задействованном узле кластера запускает BOINC-менеджер.
5. На главном узле кластера периодически запускается специальный скрипт, который анализирует очередь кластера.
6. Логика работы скрипта состоит в том, что BOINC-задачи не должны мешать запуску заданий других пользователей кластера. Если в очереди кластера появляются задания, которые могли бы запуститься после остановки BOINC-задач, то требуемое количество BOINC-задач прерывается. Если есть свободные узлы, и при этом нет незапущенных заданий в очереди, то скрипт запускает BOINC-задачи на узлах с помощью MPI-программы.

Скрипт CluBORun оперирует с очередью заданий кластера с помощью стандартных команд (запуска MPI-задачи, остановки MPI-задачи, получения данных о текущем состоянии очереди). При этом у разных планировщиков

заданий могут быть разные имена команд, а также разная специфика их работы. Поэтому для каждого планировщика необходимо разрабатывать отдельную версию скрипта CluBORun. В [4] были описаны реализации CluBORun для двух планировщиков заданий кластеров – Cleo и СУППЗ. Позднее (см. [5]) в CluBORun была добавлена возможность работы с планировщиком SLURM.

Далее описаны основные изменения, сделанные в CluBORun после выхода статьи [5]. Во-первых, была сделана реализация CluBORun для работы с планировщиком PBS Torque. Отметим, что PBS Torque (как и SLURM), является одним из самых популярных планировщиков заданий кластеров в мире. Ресурсы одного из кластеров, работающего под управлением PBS Torque, были успешно использованы в проекте SAT@home для решения ряда ослабленных задач криптоанализа поточного шифра Vivium [6], а также для нахождения новых пар ортогональных диагональных латинских квадратов порядка 10 [7].

Второе изменение изменение коснулось используемой версии VOINC. Изначально в рамках CluBORun использовался VOINC версии 6.12.22. Данная версия, в которой реализован только интерфейс командной строки, рекомендована на официальном сайте VOINC [8] для использования на старых x86-совместимых операционных системах (ОС) семейства Linux. В 2013 году мы выбрали именно эту версию VOINC, т.к. она запускалась на всех кластерах, к которым на тот момент у нас был доступ. Но для запуска и этой версии на кластерах пришлось приложить дополнительные усилия. Все выложенные сборки VOINC для ОС семейства Linux являются динамическими, а часть требуемых для запуска VOINC библиотек (например, libcurl), по умолчанию отсутствовали на всех используемых нами кластерах. Как было отмечено ранее, одна из основных особенностей CluBORun состоит в том, что для его работы на кластере не нужны администраторские права. Поэтому все действия, которые мы предпринимаем на кластере, производятся в рамках обычных пользовательских прав. Для решения упомянутой выше проблемы были использованы локальные версии требуемых библиотек, а VOINC был настроен на работу с ними с помощью символических ссылок. В результате VOINC версии 6.12.22 был успешно запущен на всех кластерах, с которыми мы работали. Но через некоторое время оказалось, что в версии 6.12.22 есть критическая ошибка, из-за которой в ряде случаев для расчета задания выбирается неверная версия клиентского приложения. Поэтому было принято решение использовать более позднюю версию VOINC, в которой эта ошибка была исправлена. После ряда консультаций с разработчиками VOINC мы сделали статическую сборку VOINC версии 7.0.65 на ОС CentOS 6.1. Версия 6.1 ОС CentOS была выбрана потому, что в ней библиотека glibc старше, чем на любом кластере, с которым мы работали. Начиная с 2015 года данная сборка VOINC успешно работает на всех кластерах, к которым у нас есть доступ. Упомянутая выше ошибка на этой версии VOINC не возникает. Еще одно преимущество данной сборки VOINC состоит в том, что теперь нам не нужно использовать локальные библиотеки на кластере – все нужные библиотеки находятся в самом исполняемом файле.

Третье изменение касается интернет-соединения между вычислительными узлами кластера и серверами проектов добровольных распределенных вычислений. До 2015 года на всех кластерах, с которыми мы работали, был открыт доступ в сеть интернет не только с главного узла, но и с каждого вычислительного узла. Но в 2015 году на одном из кластеров, на котором нами был запущен CluBORun, заблокировали доступ в сеть интернет с вычислительных узлов. Поэтому мы стали использовать прокси-сервер, настроенный на управляющем узле кластера. Конкретно, мы использовали для этих целей tinurgho [9]. Прокси-сервер tinurgho необходимо настроить один раз на управляющем узле кластера. Также для каждого BOINC-менеджера в файле `cc_config.xml` (это стандартный конфигурационный файл BOINC) необходимо указать порт и ip-адрес прокси-сервера, запущенного на управляющем узле кластера. После этого BOINC-менеджер на каждом вычислительном узле будет устанавливать интернет-соединение с сервером проекта добровольных вычислений через прокси-сервер управляющего узла.

Отметим, что все описанные изменения в работе CluBORun были нами сделаны в ситуации, когда у нас был доступ к кластеру, на котором по некоторой причине либо CluBORun не мог быть запущен, либо была невозможна корректная работа самого BOINC. В дальнейшем мы планируем модифицировать CluBORun для работы на кластерах, на которых не разрешается запуск пользователями прокси-серверов на управляющем узле. Для решения этой проблемы подходит ssh-туннелирование, использованное, например, в системе VNB-Grid [10]. Также мы планируем сделать реализации CluBORun для некоторых новых планировщиков заданий кластеров.

Работа получила частичную поддержку РФФИ (гранты 14-07-00403-А и 15-07-07891-А) и Совета по грантам для молодых ученых при Президенте РФ (стипендия СП-1184.2015.5, грант ВШШ-5007.2014.9).

### Список литературы

1. Посыпкин М.А. Грид-системы из персональных компьютеров в России: текущее состояние и перспективы // Сборник избранных трудов VII Международной научно-практической конференции «Современные информационные технологии и ИТ-образование». Москва. Изд-во Интуит.ру. 2012.
2. Ватулин Э.И. Добровольный метакомпьютинг: современное состояние и перспективы развития // Сборник материалов IX Международной конференции «Распознавание 2010». Курск. Изд-во ЮЗГУ. 2010.
3. Заикин О.С., Семенов А.А., Посыпкин М.А. Процедуры построения декомпозиционных множеств для распределенного решения SAT-задач в проекте добровольных вычислений SAT@home // Управление большими системами. 2013. Вып. 43. С. 138-156.
4. Манзюк М.О., Заикин О.С., Посыпкин М.А. CluBORun: программный комплекс для использования свободных ресурсов вычислительных кластеров в BOINC-расчетах // Информационные технологии и вычислительные системы. 2014. № 4. С. 3-11.

5. Afanasiev A.P., Bychkov I.V., Manzyuk M.O., Posypkin M.A., Semenov A.A., Zaikin O.S.. Technology for integrating idle computing cluster resources into volunteer computing projects // Proceedings of The 5th International Workshop on Computer Science and Engineering: Information Processing and Control Engineering (WCSE 2015-IPCE). 2015, pp. 109-114.
6. Semenov A., Zaikin O. Using Monte Carlo method for searching partitionings of hard variants of Boolean satisfiability problem // In: Malyshkin, V. (ed.) Parallel Computing Technologies - 13th International Conference, PaCT 2015, Petrozavodsk, Russia, August 31 - September 4, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9251, pp. 222–230. Springer (2015).
7. Заикин О.С., Кочемазов С.Е. Поиск пар ортогональных диагональных латинских квадратов порядка 10 в проекте добровольных распределенных вычислений SAT@home // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2015. Т.4. № 3. С. 95-108.
8. Доступные для загрузки версии BOINC. URL: [https://boinc.berkeley.edu/download\\_all.php](https://boinc.berkeley.edu/download_all.php) (дата обращения 19.10.2015).
9. Прокси Tinyпроху. URL: <https://banu.com/tinyпроху/> (дата обращения 19.10.2015).
10. Evtushenko Y., Posypkin M., Sigal I. A framework for parallel large-scale global optimization // Computer Science - Research and Development. 2009. Vol. 23. Issue 3. pp. 211-215.