
BOINC-сервер

[Установка BOINC-сервера](#)

[Создание и удаление проектов](#)

[Запуск контейнеров с BOINC-сервером](#)

[BOINC FAQ](#)

BOINC-клиент

[Установка BOINC-клиента](#)

[Управление BOINC-клиентом \(справочник команд\)](#)

Актуальные задачи по разработке BOINC

[In English](#)

1. Эффективные способы управления заданиями (на стороне сервера или клиента), много предложений в архивах списков рассылки BOINC).
 - использовать в нынешней системе управления заданиями среднее квадратическое отклонение вместо среднего значения;
 - на стороне клиента - отслеживать и учитывать время загрузки файлов
 - на стороне сервера - реализовать возможность присылать клиенту мелкие задания, даже если он набрал больших
 1. Исследовать на стороне клиента, какие приложения эффективно работают вместе во многопроцессорном режиме (есть наблюдение, что из-за особенностей обращения к памяти некоторые комбинации приложений работают эффективнее других). Оценить, стоит ли реализовать эти наблюдения для распределения заданий. Если стоит, то реализовать.
 2. Вычисления, интенсивные по передаче данных.
 - выполнять на одном и том же клиенте задания с общими «промежуточными» файлами
 - потоковые вычисления (например, см. IBM Infosphere)
 - модели вычислений на большом статическом множестве данных (например, см. MapReduce, веб-сервис Amazon EC2)
 - оценка и использование физической близости клиентов
 - передача данных напрямую между клиентами
 1. Виртуальная машина на стороне клиента (<http://boinc.berkeley.edu/trac/wiki/VmApps>).
 - «Volunteer cloud» для ученых (есть совместный проект CERN и INRIA)
-

-
1. Адаптивная репликация. В BOINC реализован эвристический алгоритм назначения разных уровней репликации в зависимости от надежности клиентов. Подумать над альтернативами, оценить эффективность того, что есть, оптимизировать.
 2. В BOINC реализован механизм защиты от 'cherry picking', когда клиенты выполняют только маленькие задания, чтобы получать больше кредитов. Исследовать его эффективность.
 3. Исследовать (!) и оптимизировать способы разделения ресурсов между проектами. (INRIA, Arnaud Legrand)
 4. Обобщить систему кредитов за вычисления.
 - назначать кредиты не только за время ЦП, но и за место на диске и др. ресурсы. Здесь теоретически была бы эффективна игровая модель.
 1. Априорные оценки длительности выполнения батчей заданий. Эффективное управление батчами заданий.
 2. Квотирование в зонтичных проектах.
 3. Реализовать систему распределенного хранения данных на основе BOINC.
 4. Оптимизировать вычисления на GPU, чтобы они работали и в фоновом режиме.
 5. Включить GPU и многопоточные приложения в механизм homogeneous redundancy (<http://boinc.berkeley.edu/trac/wiki/HomogeneousRedundancy>).
 6. «Микрозадания» (не поняла смысл).
 7. Исследовать возможности подключения BOINC к GRID и облакам.
 8. Эффективные способы сбора статистики доступности клиента.
-