

**10th International Conference "Distributed Computing and Grid
Technologies in Science and Education" (GRID'2023),
Dubna, Russia, July 3-7 2023**

Simulation of voluntary distributed computing projects using the ComBoS simulator

Ilya I. Kurochkin

IITP RAS, Moscow, Russia

**Моделирование проектов добровольных
распределенных вычислений
с помощью симулятора Combos**

Курочкин И.И.

ЦРВ ИППИ РАН, Москва, Россия

Особенности грид-систем

- Гетерогенность узлов распределенной системы;
- Автономность расчетов на различных узлах и невозможность постоянной координации расчетов между узлами;
- Ненадежность связей и возможное отключение вычислительных узлов;
- Непостоянное время непрерывной работы узла и трудность расчета длительных заданий;
- Наличие ошибок и задержек при расчетах.

Платформы для организации распределенных вычислений

- HTCondor
- Globus
- BOINC
- Oracle Grid Engine



Платформа BOINC



BOINC – Berkeley Open Infrastructure for Network Computing
Платформа для организации добровольных распределенных вычислений:

- состоит из серверной и клиентской части;
- дает возможность задействовать вычислительные мощности персональных компьютеров(ПК);
- кроссплатформенная клиентская часть;
- гибкая настройка клиентской части для эффективного использования ресурсов ПК.

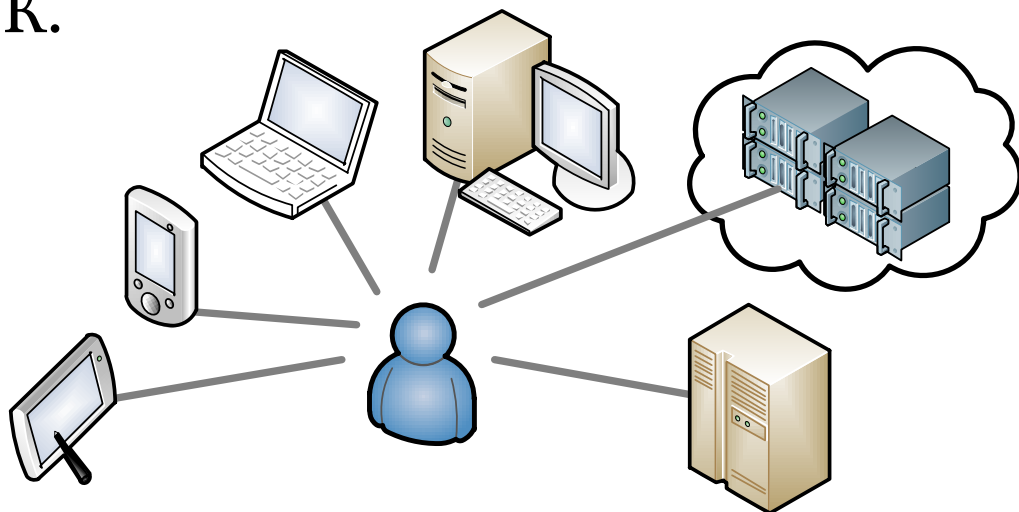
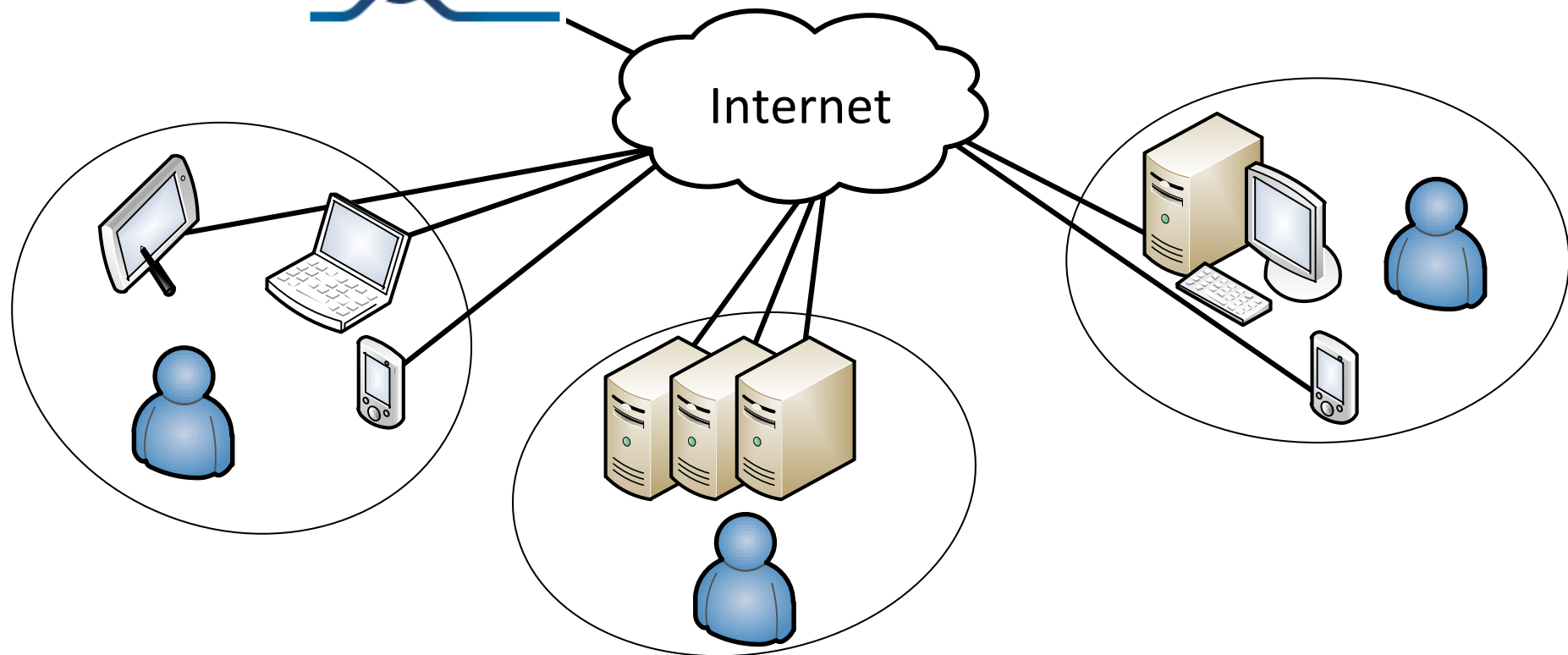


Схема подключения вычислительных узлов к грид-системе (проекту распределенных вычислений, ДРВ)



Грид-системы из персональных компьютеров

Развитие ГСПК стало возможным из-за появления у рядового пользователя достаточно мощных вычислительных устройств и увеличения доступности к глобальной сети Интернет.

Одной из основных причин появления ГСПК является использование простаивающих вычислительных ресурсов, так у большинства пользователей персональных компьютеров уровень использования находится на уровне 10-15%.

Одним из подвидов грид-систем из персональных компьютеров (ГСПК) являются публичные ГСПК, в которых вычислительные мощности предоставляются добровольцами.

Такие публичные ГСПК называют проектами добровольных распределенных вычислений (ДРВ) и часто дают им название по научной проблеме или вычислительной задаче, которая решается в рамках проекта, к примеру, SAT@home, NetMax@home, LHC@home, Asteroids@home.

Размер таких ГСПК может быть большим и достигать миллионов вычислительных узлов, как в проекте Folding@home.



Российские проекты ДРВ

- SAT@home (криптография, SAT-подход)
- Optima@home (решение задач конечномерной оптимизации)
- **Parlea.ru** (студенческий зонтичный проект)
- NetMax@home (имитационное моделирование телекоммуникационных сетей)
- **USPEX@home** (поиск новых материалов)
- Acoustics@home (исследование дна Японского моря)
- AndersonAttack@home (криптография)
- **Gerasim@home** (комбинаторика, латинские квадраты)
- XANSONS for COD (материаловедение)
- **RakeSearch** (поиск латинских квадратов с опр. свойствами)
- ODLK@home (поиск канонических форм латинских квадратов)
- **Amicable Numbers** (поиск дружественных чисел)
- **SiDock@home** (поиск лекарств от COVID-19)
совместный проект (Россия-Словения)

Тип задачи «bag of tasks»

Задача, которая может разбиваться на множество независимых подзадач. В процессе расчета такой подзадачи не нужен обмен информацией с другими подзадачами или с координатором расчетов (сервером). Каждая подзадача будет рассчитываться на отдельном вычислительном узле распределенной системы.

Как правило, для различных подзадач используются различные наборы входных данных и единый алгоритм их обработки.

Такой тип задач в литературе называется «bag of tasks» или **задача, разделяемая по данным**.

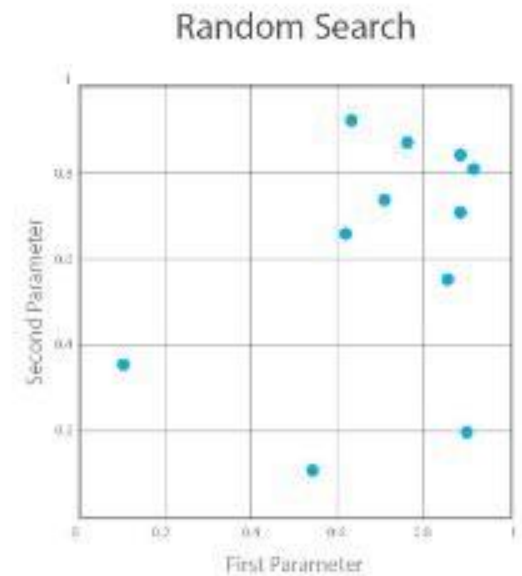
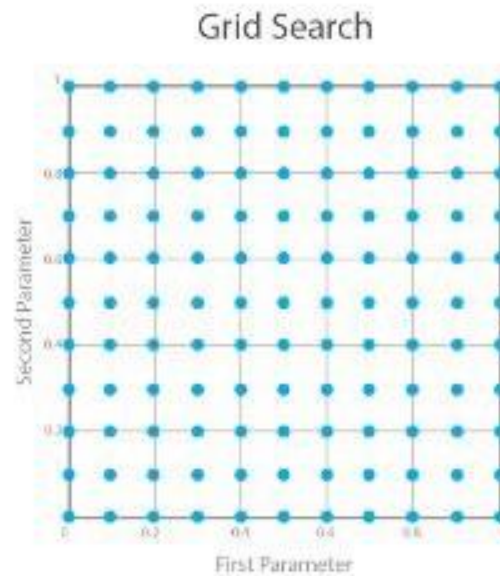
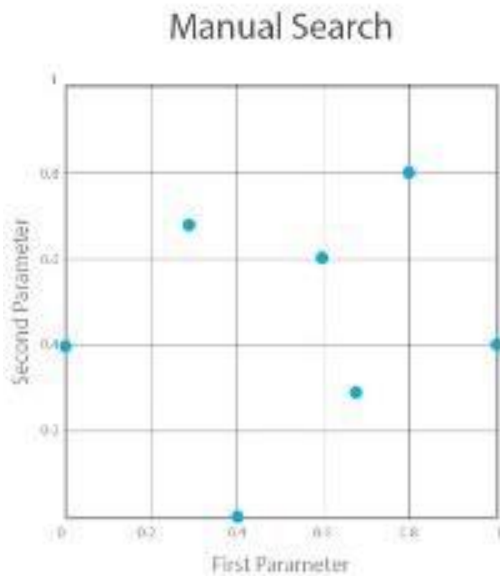
В качестве примера таких задач можно привести задачи:

- Задачи комбинаторики и полного перебора;
- Задачи обработки изображений и других мультимедийных данных;
- задачи имитационного математического моделирования и др.

Задачи полного перебора

- Известны сложность и количество подзадач
- Тип “Bag of Tasks”
- Асинхронное выполнение

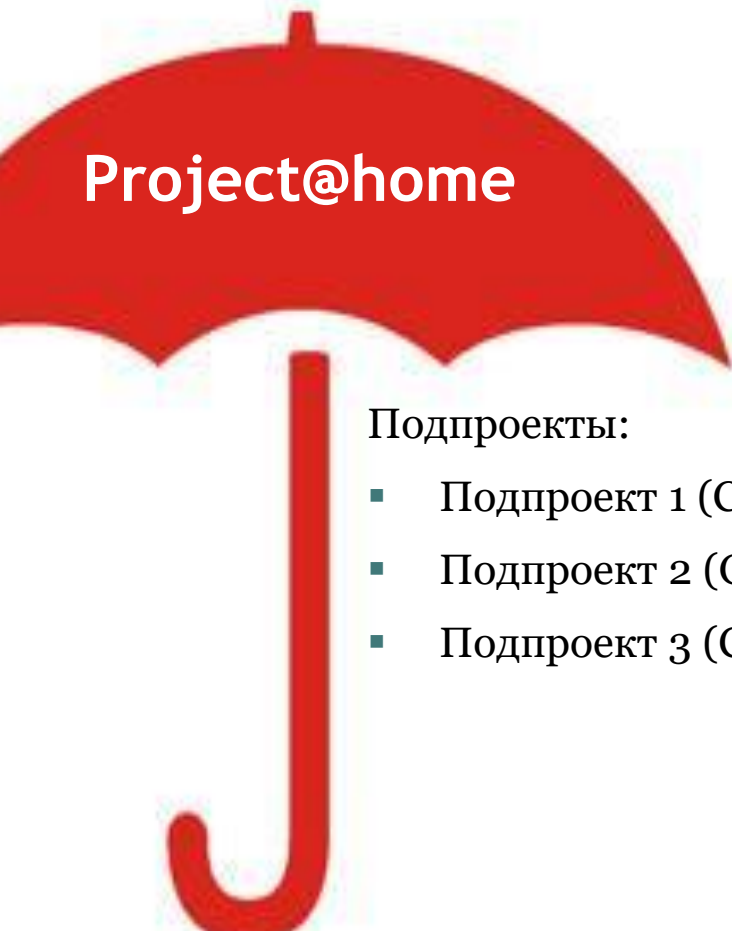
Пример: grid search



Комбинаторные задачи

- Сложность каждой подзадачи может быть разной (иногда непрогнозируемой)
- Количество подзадач известно, но могут быть изменения
- Тип “Bag of Tasks”
- Асинхронное выполнение, но необходимы сроки выполнения (deadline)

Зонтичный проект распределенных вычислений

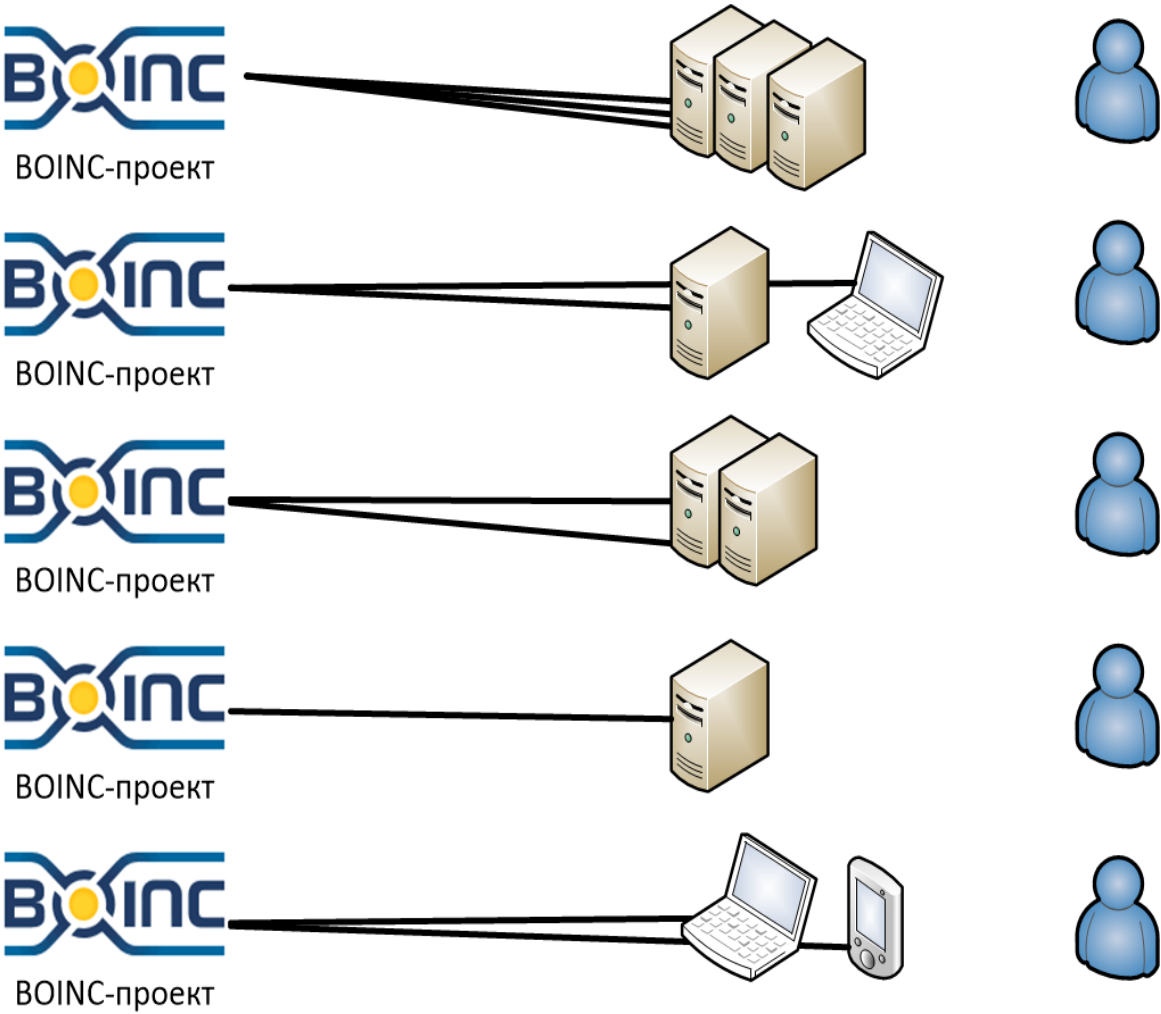


- Возможность запускать небольшие эксперименты
- Нет необходимости разворачивания нового проекта
- Не нужно развивать проект с нуля
- Существенная вычислительная мощность на старте эксперимента

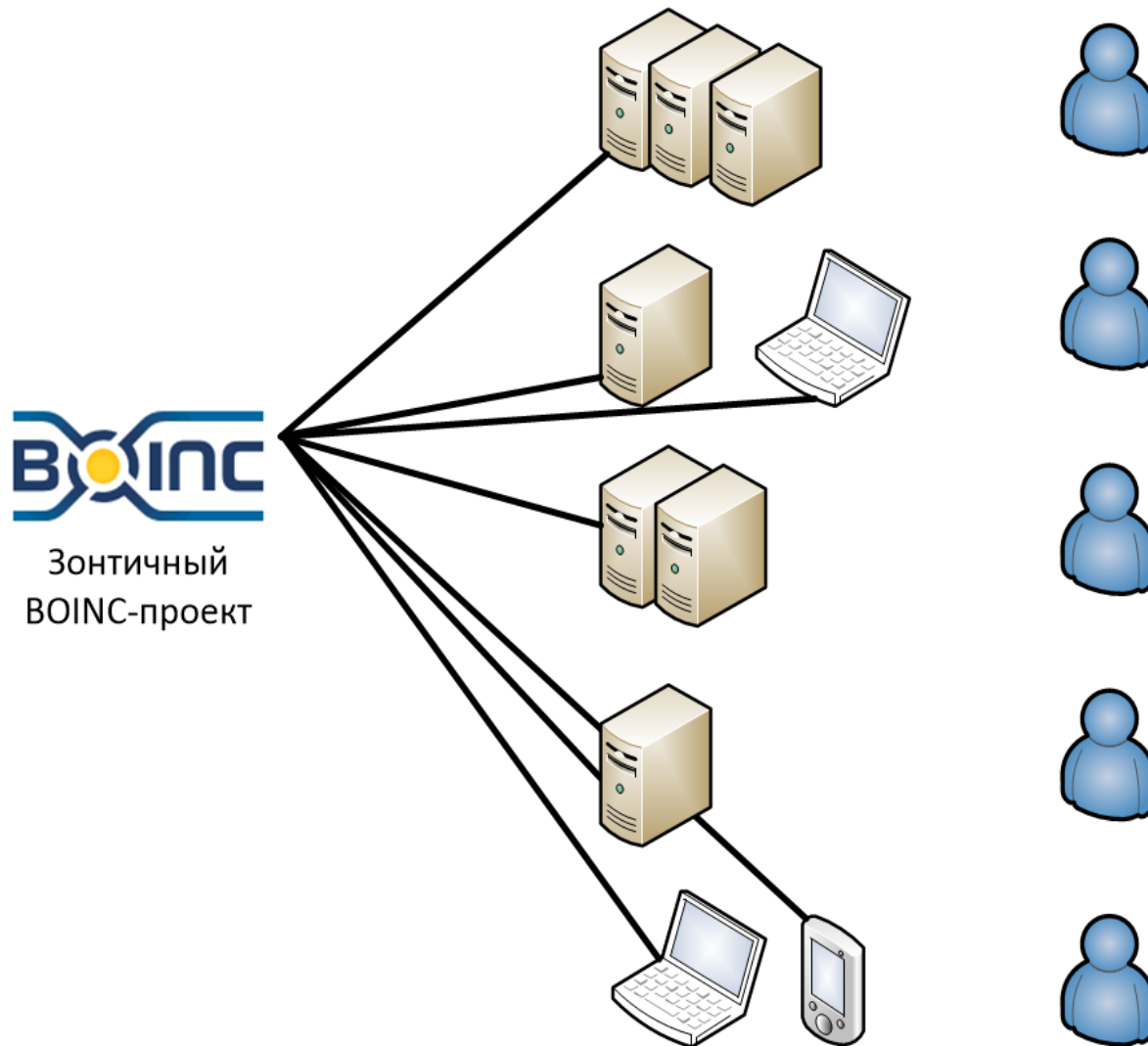
Подпроекты:

- Подпроект 1 (CPU)
- Подпроект 2 (GPU)
- Подпроект 3 (CPU + GPU)

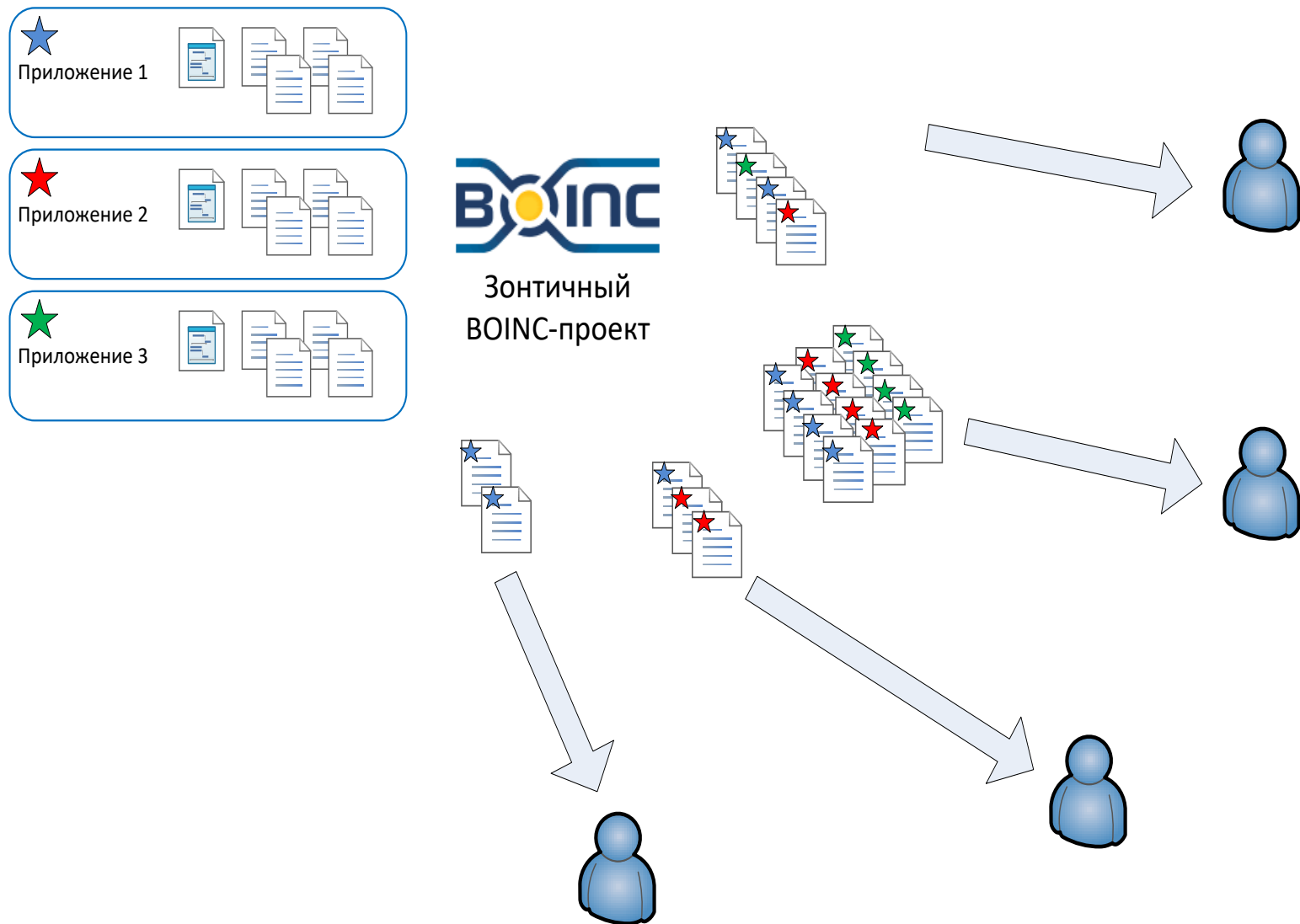
Плохой вариант организации BOINC-проектов для нескольких экспериментов



Правильный вариант организации BOINC-проектов



Распределение заданий в зонтичном проекте



Повышение вычислительной мощности проекта ДРВ

- **Технические методы**

- Разработка версий вычислительного приложения под различные операционные системы и архитектуры CPU/GPU
- Тонкая настройка параметров проекта
- Обеспечение целостности данных
- Увеличение отказоустойчивости сервера проекта
- Доработка системы начисления баллов

- **Социологические подходы**

- Привлечение новых добровольцев
- Удержание интереса добровольцев

Инструменты имитационного моделирования грид-систем

- SimGrid
- GridSim
- EmBOINC
- The BOINC Client Emulator
- SimBA
- SimBOINC
- MicroGrid, MobiNet (эмуляторы)
- **ComBos** (на основе SimGrid)



The BOINC Client Emulator



Особенности балансировки нагрузки (выдачи заданий) в зонтичном проекте ДРВ

- Запрос заданий клиентами (вычислительными узлами грид-системы)
- Динамические ограничения использования ресурсов клиентами
- Различное (иногда непрогнозируемое) время выполнения различных вычислительных приложений
- Асинхронное выполнение различных вычислительных приложений
- Разные режимы работы и разная надежность вычислительных узлов

Эксперименты по имитационному моделированию

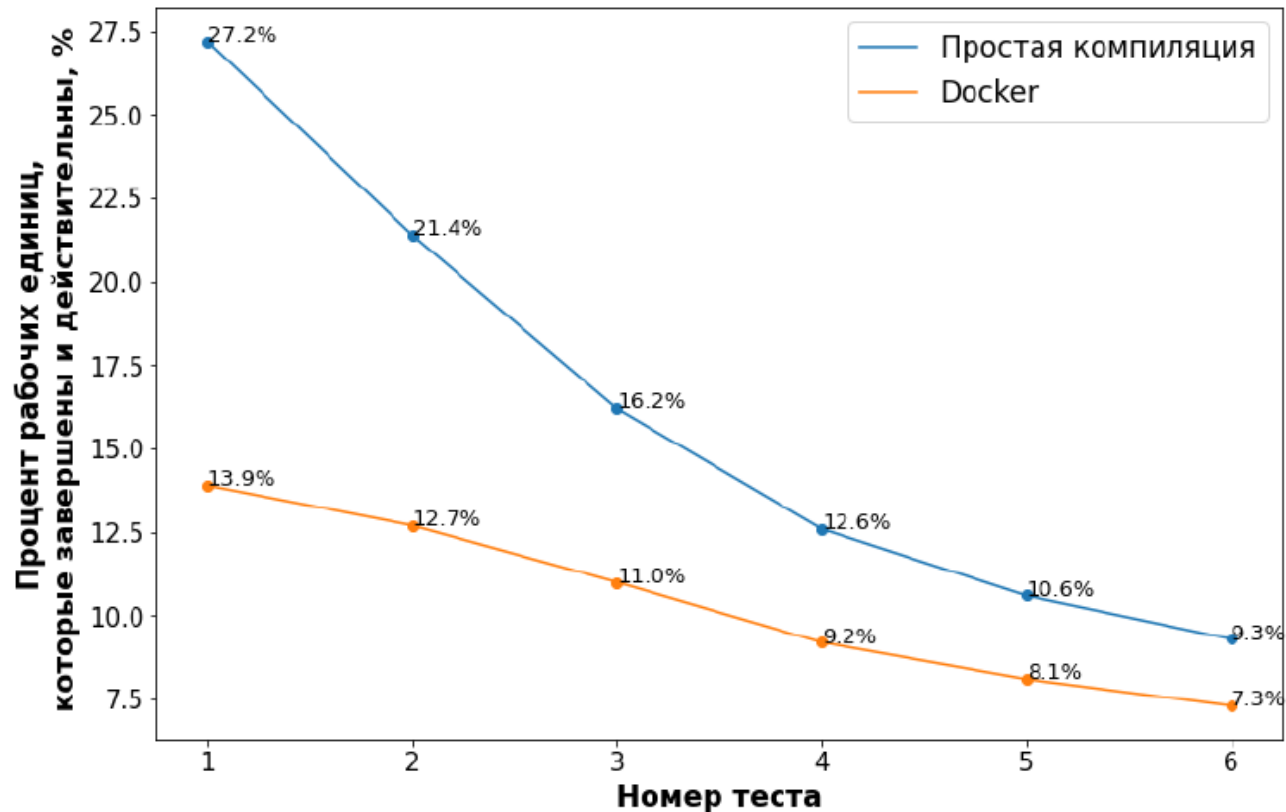
- Исследование статистических методов балансировки нагрузки
- Адаптация методов балансировки нагрузки для других распределенных систем (к примеру, метод Коновалова-Разумчика)
- Исследование различных типов запуска вычислительных приложений

Сравнение различных типов запуска вычислительного приложения

	Размер входного файла в Mbytes	Время выполнения в GFLOPS	Размер выходного файла в Mbytes	Время вып. в мин/ Время скач. файлов
Легкий тест				
local	150	1900	10	~6m20s/32s
docker	350	2000	10	~6m40s/1m2s
Средний тест				
local	250	3900	10	~13m/52s
docker	450	4000	10	~13m20s/1m32s
Тяжелый тест				
local	550	4900	10	~16m20s/1m52s
docker	750	5000	10	~16m40s/2m12s

Оценка длительности выполнения задания и транспортировки входных и выходных файлов, при условии скорости передачи 40 Mbit/s и средней мощности вычислительного узла 5 GFLOPS/s

Сравнение различных типов запуска вычислительного приложения



Результаты симуляций с изменяемым размером входных файлов и фиксированным временем выполнения программ при использовании сборки с Docker и без него

Эксперименты по тестированию алгоритмов балансировки нагрузки

- 2 проекта с успешностью выполнения 95% и 50%
- 150 клиентов

Метод	Проект 1	Проект 2	Суммарное число задач
Baseline	54,170	13,038	67208
MAX <i>suc_rate</i>	54,175	13,155	67330 (+0.18%)
WRR	54,152	13,113	67285 (+0.1%)

Заключение

- ComBoS может быть успешно использован для моделирования грид-системы на платформе BOINC;
- Возможность доработки данного симулятора:
 - Для специфических вычислительных приложений;
 - Для симуляции использования контейнеров docker;
 - Для зонтичных проектов ДРВ.
- ComBoS позволяет протестировать алгоритмы балансировки нагрузки

Спасибо за внимание

Благодарности:

Леванков Егор (бакалавр ВШЭ)

Акимов Александр (бакалавр ВШЭ)

Шакиров Айвар (бакалавр ВШЭ)

Папанов Валерий (бакалавр НИТУ МИСИС)

Долгов Андрей (студент магистратуры НИТУ МИСИС)

Центр распределенных вычислений

ИППИ РАН

web: distributed-computing.ru

e-mail: kurochkin@iitp.ru