# Применение генетических алгоритмов для планирования заданий в GRID

**Сапронов Андрей Юрьевич**
**Шаповалов Тарас Сергеевич**

Вычислительный Центр
Дальневосточного отделения РАН

# GRID

**Основные свойства GRID имеющих важное значения для постановки задач планирования и распределения ресурсов:**

- Многокомпонентность
- Распределенность
- Гетерогенность (архитектуры и программного обеспечения)
- Динамичность топологии

# Запуск заданий

**Два подхода к планирование заданий:**

- после постановки задания на счет оно сразу же запускается на оптимальных для задачи вычислительных ресурсах
- предварительное составление некоторых планов обработки заданий – так называемых расписаний

# Формальная постановка задачи

Множество задач $U = \{u_0, ..., u_N\}$ на отрезке планирования $[0, T]$

$u_o$ - холостая задача (для обозначения простаивания ресурса

Множество $S = \{s_0, ..., s_M\}$

$$s = \left\{ (\alpha_i,\ \beta_i),\quad i = \overline{1, N},\quad \alpha_i,\ \beta_i \in [0, T] \right\}$$

Наличие доступных ресурсов с течением времени: $r_t = \left\{ r_t^j,\quad j = \overline{1, J} \right\},\quad t \in [0, T]$

$J$ - число типов ресурсов

# Формальная постановка задачи

Условия допустимости расписания:

$$\forall u_i \in U: \quad \beta_i \le T,$$

$$\forall u_i, u_j \in U: \quad u_i \prec u_j \Rightarrow \beta_i \le \alpha_j,$$

$$\forall t \in [0,T], U_t = \{u_i \in U \mid \alpha_i \le t \le \beta_i\}: \quad r^j(t) = \sum_{u_i \in U_t} r_{ij} \le r_t^j,$$

$U_t$ - множество задач, выполняемых в момент времени $t$;

$r^j(t)$- потребность в ресурсе $j$-го типа, $r_t^j$ - наличие такого типа ресурсов

Для $i$-й задачи задается длительность $\tau_{ij} > 0$ её выполнения на $j$-м ресурсе

Основной критерий оценки – длина расписания:

$$l(s) = \max_{1 \le i \le N} \beta_i$$

Лучшее расписание по данному критерию будет следующим:

$$l(s') = \min_{1 \le j \le M} \left\{ \max_{1 \le i \le N} \beta_i \right\}$$

# Существующие методы решения

- Точные методы (гарантируют нахождение расписания)
  - Линейного и целочисленного программирования
  - Ограниченного перебора
  - Метод критического пути
- Эвристические методы (не гарантируют лучшее решение)
  - Использующие эвристики планирования
  - Использующие эвристики упорядочения
  - Иерархический подход
  - Имитация отжига
  - Подходы, основанные на теории искусственного интеллекта
    - Экспертные системы
    - Системы основанные на знаниях (knowledge-based)
  - Эволюционные алгоритмы (генетический алгоритм)

# Задача в терминах ГА

•Массивы значений определяющих расписания, кодируются одной строкой. Каждая такая строка в терминах ГА является **особью**
• Множество таких  строк составляют **популяцию** возможных расписаний для GRID.
•Особи классифицируются с помощью целевой функции (**функции пригодности**)

**Пример:**
Можно выбрать форму **хромосомы**, состоящую из нескольких блоков. Каждый из блоков представляет собой описание выполнения задач на одном процессоре в GRID в виде пары <идентификатор задачи, требуемое процессорное время>. Такая пара в терминологии ГА является **геном**. Если в расписании есть окна, в которых тому или иному процессору приходится простаивать, то такая задача простоя имеет идентификатор ноль.

# Алгоритм

1. Формирование первой популяции (множества расписаний) случайным образом.
2. Оценка каждого расписания функцией пригодности.
3. Процесс селекции – вероятностный процесс выбора тех расписаний, которые будут участвовать в формировании следующего поколения, коррелирующий с оценками функции пригодности.
4. Применение к расписаниям генетических операторов скрещивания и мутации (с проверкой новообразовавшихся расписаний на корректность).
5. Оценка каждого расписания функцией пригодности; переход на пункт 3.
6. Процесс поиска останавливается, когда изменения значений функции пригодности от поколения к поколению становятся незначительными. В этом случае выбрать расписание с максимальным значением функции пригодности.

# Особенности алгоритма

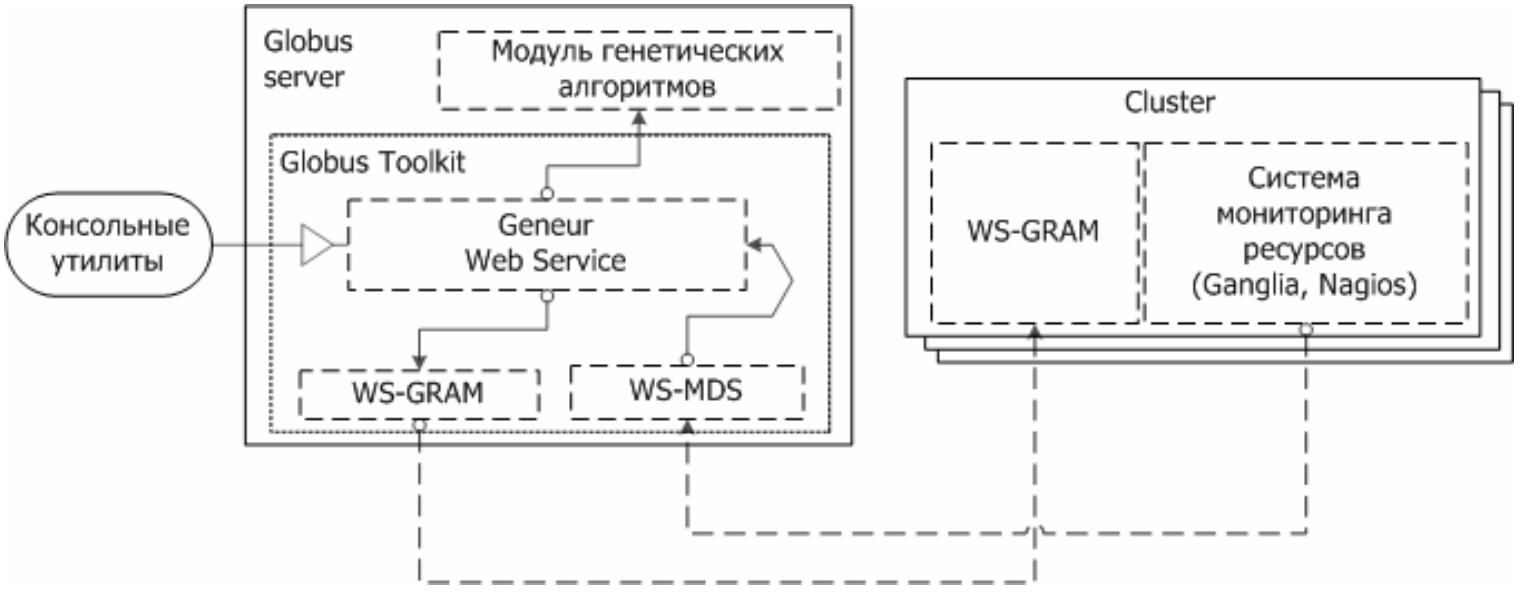
•**Процесс мутации.** Применяются два генетических оператора: кроссинговер (делении строки расписания на несколько частей в определенных точках и обмене такими частями между расписаниями) для обмена генетического материала между особями и оператор мутации с обменом ген (случайная перестановка генов в одной хромосоме)

•**Функция пригодности**

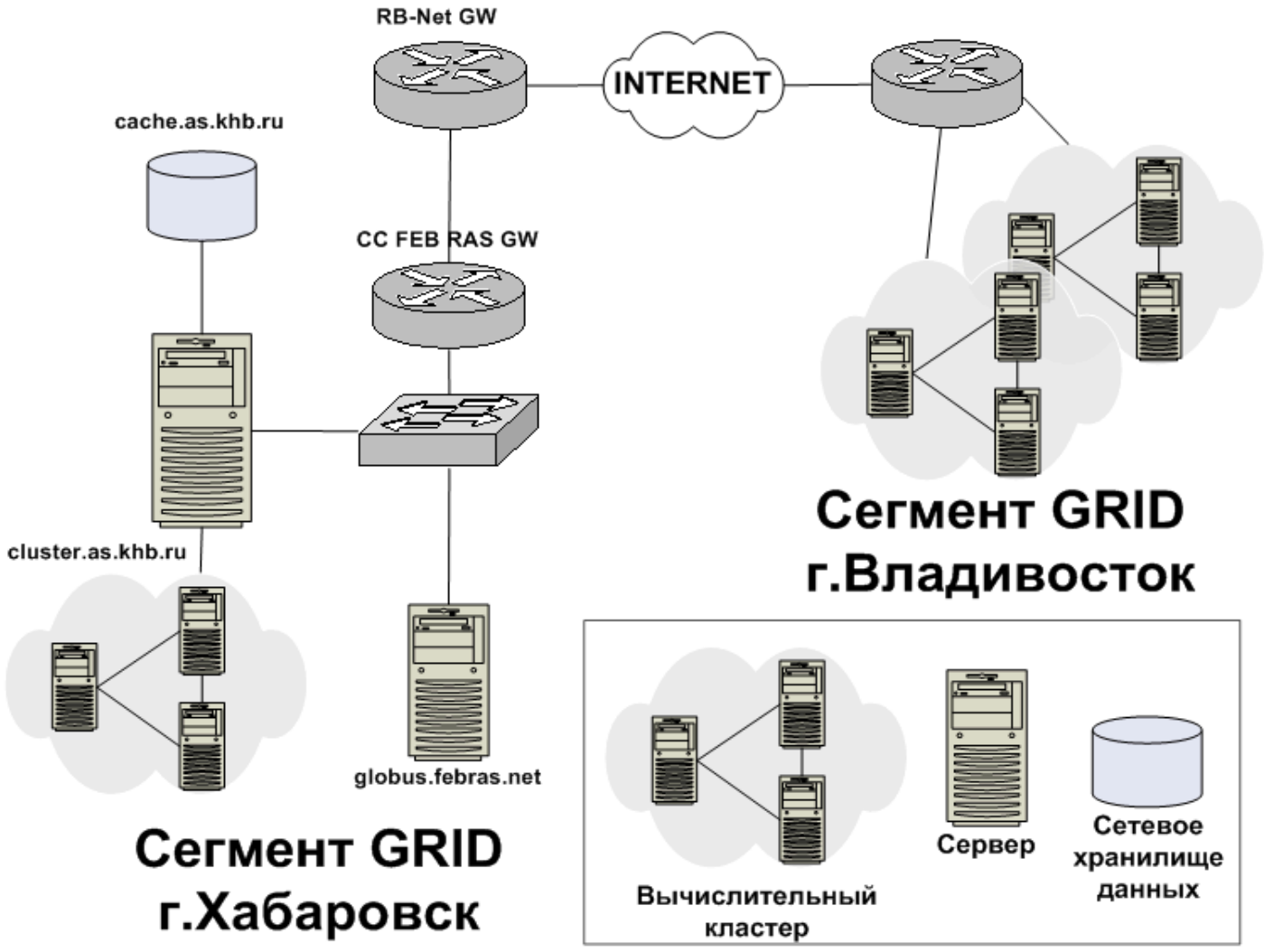
•**Принцип элитизма.** Формирование первой популяции на очередном цикле алгоритма осуществляется с использования множества наиболее  приспособленных особей полученных на предыдущем этапе.

•**Степень разнообразия популяции.** Используется ряд приемов препятствующих нахождению решения как локального максимума

# Место в инфраструктуре Globus Toolkit



# Экспериментальная база



# Результаты

**Результаты:**

- Описанный алгоритм реализован в форме соответствующих модулей для платформы Globus Toolkit. Описание заданий осуществляется на языке JSDL (Job Submission Description Language)
- Проведены исследования по изучению зависимостей алгоритма от основных параметров: вероятность применения оператора мутации, числа точек разрыва операции скрещивания, размера популяции, доля хромосом участвующих в элитизме
- Алгоритм показал хорошую скорость сходимости как в гомогенных, так и в гетерогенных средах

**Дальнейшее развитие:**

- оценка целесообразности практического использования в небольших виртуальных организациях аналогичных GRID сети ДВО РАН

# Выводы

- **Многокомпонентность:** ГА  в отличии от других методов, например методов линейного программирования или перебора, имеют линейную скорость сходимости
- **Распределенность и гетерогенность** архитектуры достаточно просто учитывается в функции пригодности
- **Динамичность топологии:** добавление или удаление вычислительных мощностей происходит без ущерба для эффективности (скорости сходимости, сложности реализации)