

Адаптация генетических алгоритмов для
выполнения в эластичной вычислительной среде
с учетом особенностей их применения в САПР.

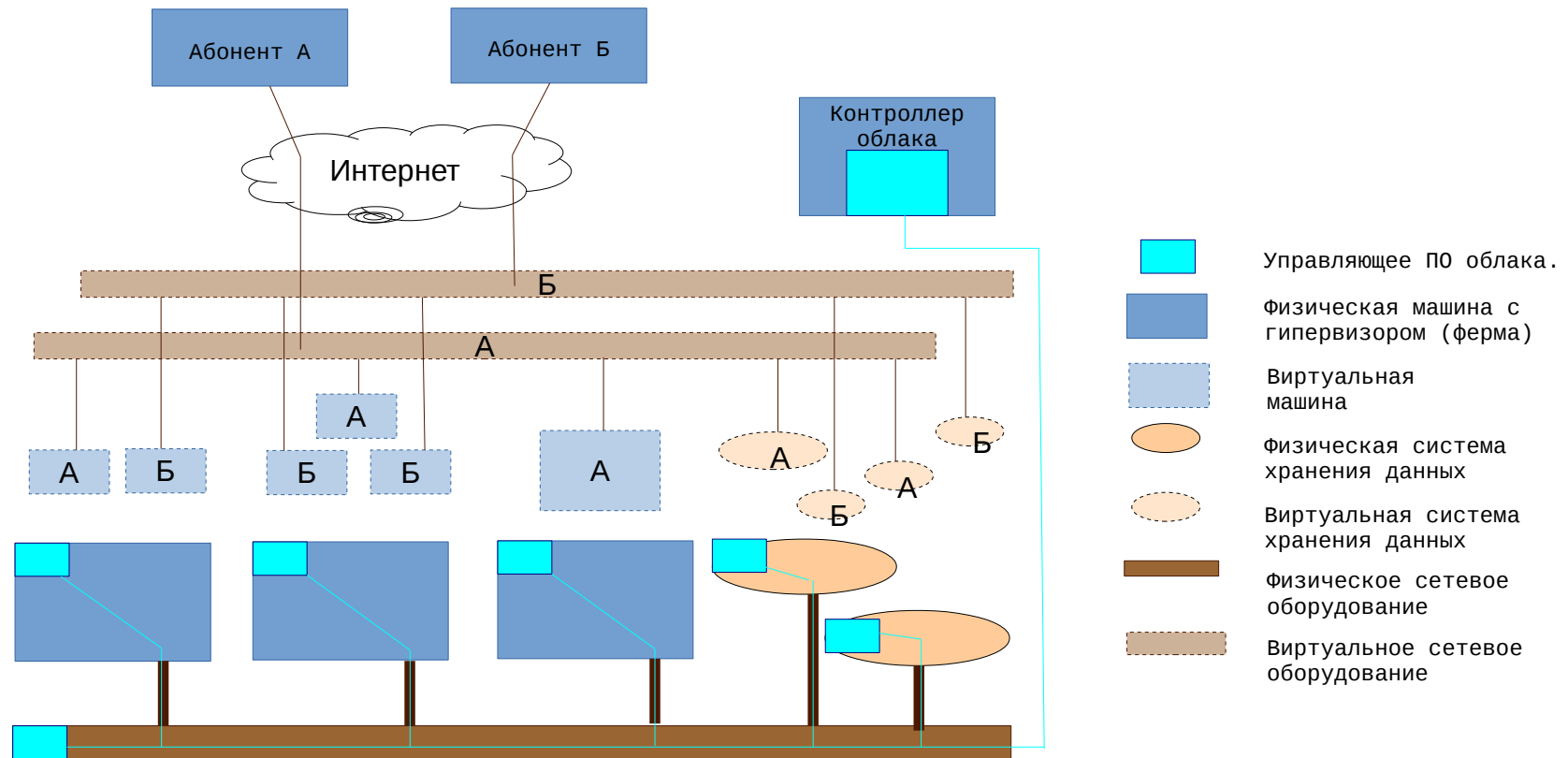
Генетические алгоритмы

- Эвристические алгоритмы поиска в пространстве решений путём случайного подбора, комбинирования и вариации искоемых параметров с использованием механизмов, аналогичных естественному отбору в природе.
- Используются для решения задач оптимизации и моделирования.
- В области САПР СБИС могут применяться на этапах синтеза и размещения блоков.
- Переборная природа ГА определяет высокие требования к вычислительным ресурсам.

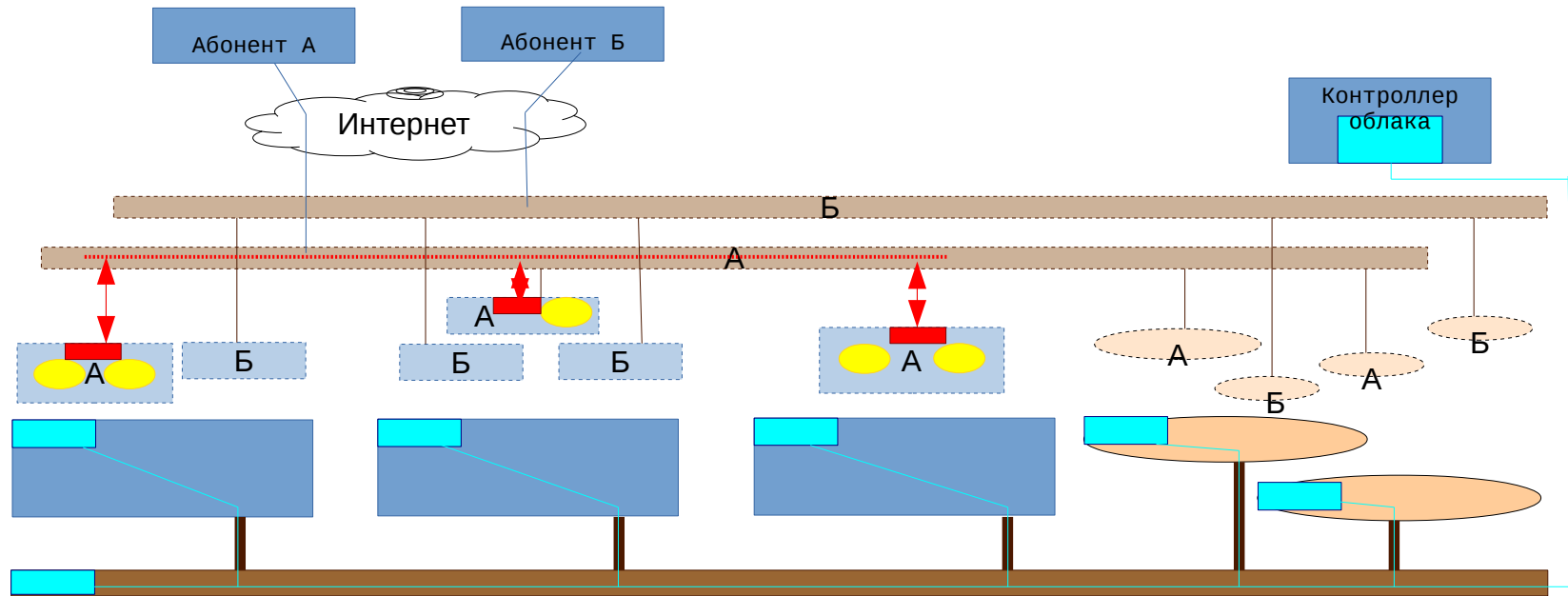
Как получить вычислительные ресурсы для своего приложения?

- Собственное оборудование - дорого, сложно
- Арендованное оборудование - неудобно
ограничения владельца
- Арендованные виртуальные машины (VDS) - может быть нужна дополнительная инфраструктура (сеть, СХД)
- Облако IaaS





Архитектура облака IaaS



Облако IaaS и HPC



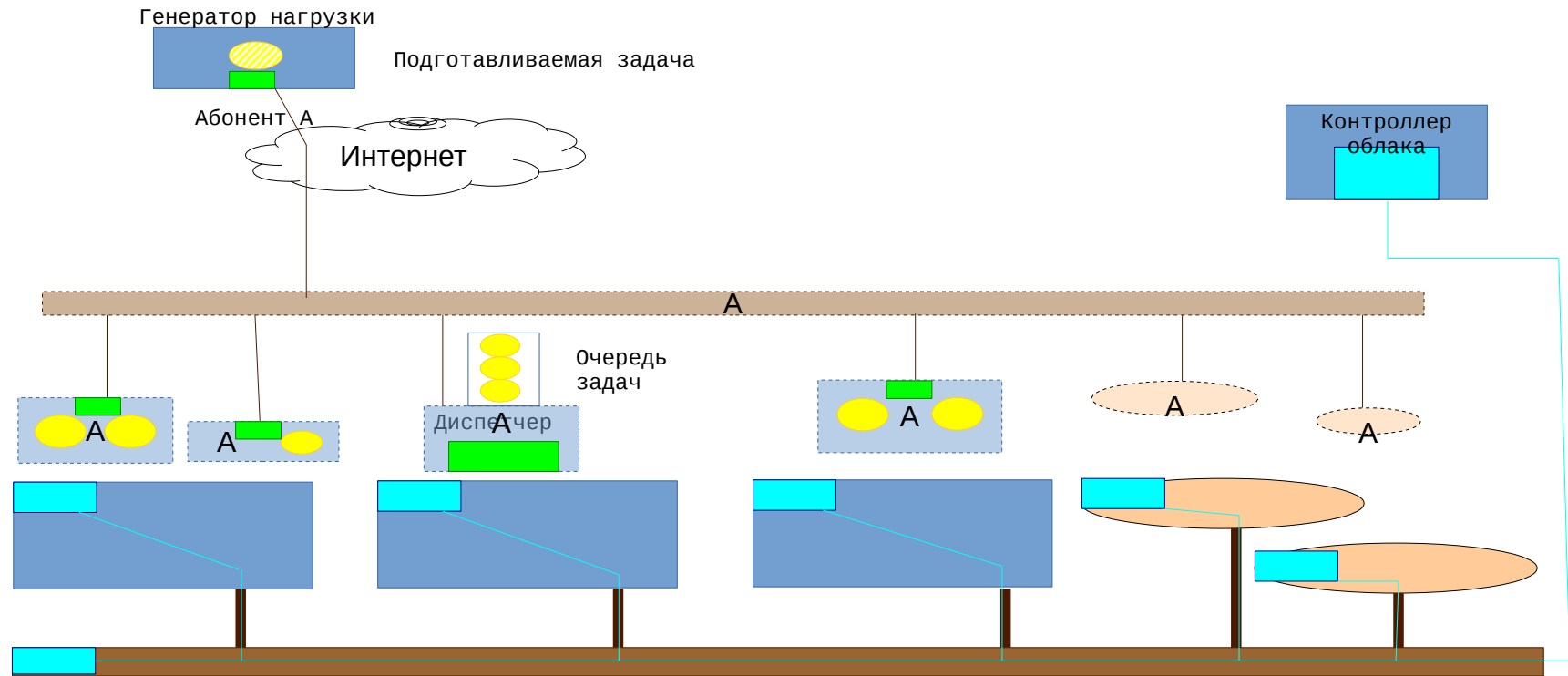
-  Управляющее ПО облака
-  Физическая машина с гипервизором (ферма)
-  Виртуальная машина
-  Физическая система хранения данных
-  Виртуальная система хранения данных
-  Физическое сетевое оборудование
-  Виртуальное сетевое оборудование

-  Распределенное приложение, запущенное в кластере из виртуальных машин.
-  Отдельный процесс распределенного приложения
-  Библиотека распределенных вычислений и драйвера высокоскоростного интерфейса
-  Имитация высокоскоростного интерфейса в сети облака



Облако и эластичность

- Эластичность заключается в автоматическом масштабировании выделяемых ресурсов в зависимости от нагрузки.
- Эластичность заявляется, как одно из ключевых свойств облачных вычислений, обеспечивающее экономическую эффективность.
- Свойством эластичности обладают приложения со специально разработанной архитектурой.
- Приложения классических параллельных архитектур (MPI, PVM) неэластичны.

IaaS + GRID



-  Управляющее ПО облака
-  Физическая машина с гипервизором (ферма)
-  Виртуальная машина
-  Физическая система хранения данных
-  Виртуальная система хранения данных
-  Физическое сетевое оборудование
-  Виртуальное сетевое оборудование

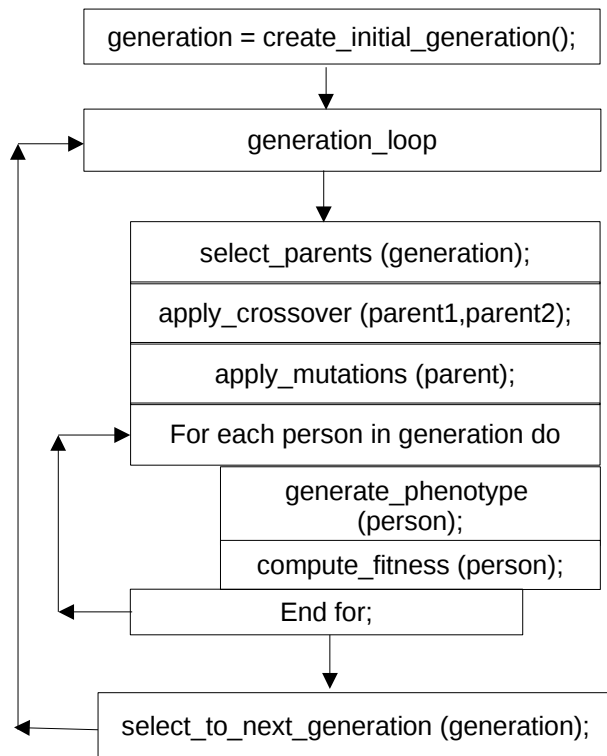
-  Управляющее ПО кластера
-  Задача

На схеме не показаны:

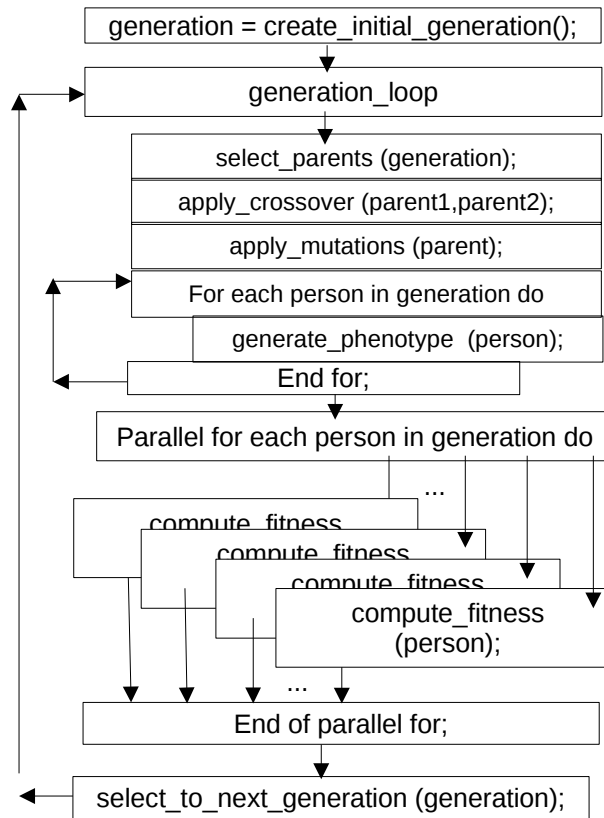
- виртуальные машины других абонентов облака
- связи между компонентами управляющего ПО кластера в виртуальной сети абонента А.

Варианты распараллеливания ГА

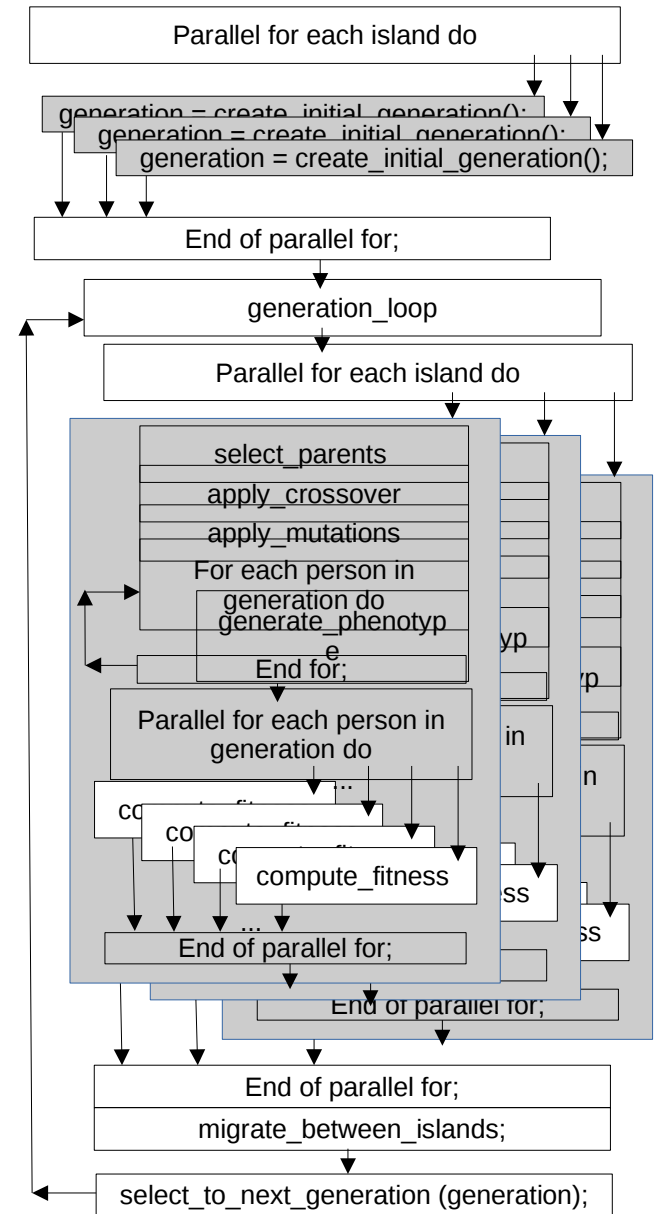
Последовательный ГА



Параллельный ГА

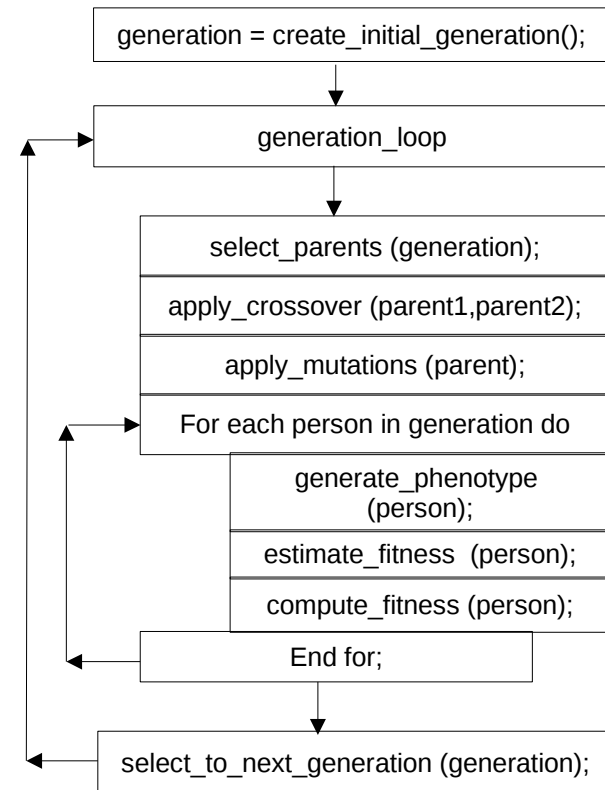


Островной параллельный ГА

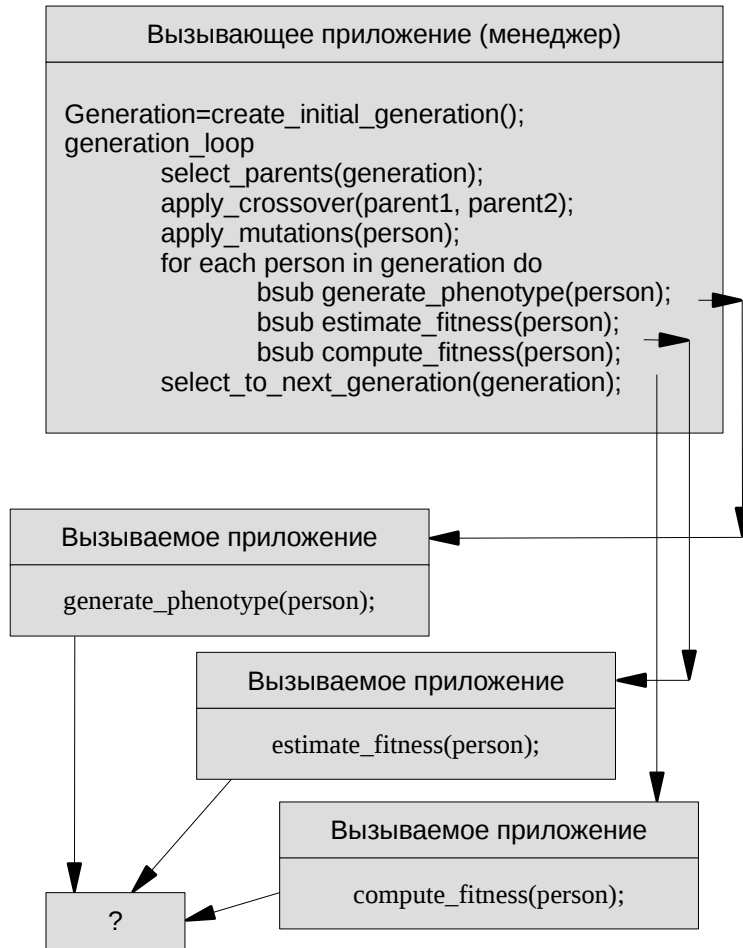


Особенности ГА применительно к САПР

- генотипом особи является описание параметров проектного решения
- для вычисления фитнес-функции могут задействоваться промышленные средства САПР, внешние по отношению к собственно генетическому алгоритму: симуляторы, трассировщики и т.д.
- как следствие - фенотипом особи является проектное решение в формате, доступном для используемых промышленных средств САПР
- как следствие — обязательно присутствует допускающий параллельную обработку этап формирования фенотипа на основе генотипа
- вычисление фитнес-функции может быть очень сложным, длительным процессом, распараллеливание которого может быть ограничено наличием лицензий на используемые промышленные средства САПР
- как следствие - может присутствовать этап грубой, но быстрой оценки фитнес-функции с помощью эвристик для отбраковывания явно дефектных особей, который также допускает распараллеливание.



Наивная реализация параллельного ГА в GRID

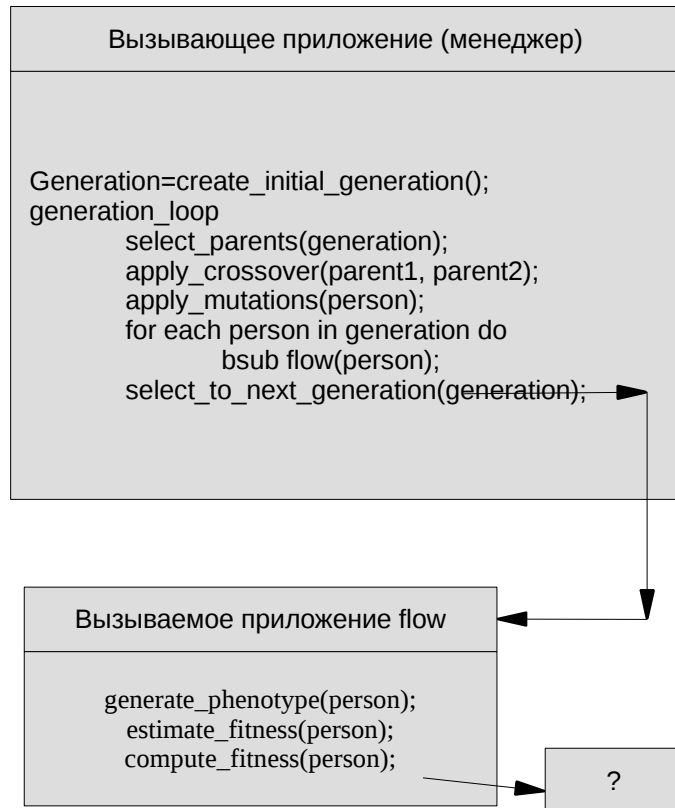


Очередность запуска приложений-подпрограмм и тем более очередность их завершения носят в значительной степени случайный характер.

Последующая часть алгоритма, например вычисление фитнес-функции для некоторой особи, может получить управление прежде, чем закончится генерация её фенотипа, что приведёт к краху алгоритма.

Приложение-менеджер должно иметь средства, чтобы избежать ситуации гонок и обеспечить контроль на возвратом управления от параллельной части к

Маршрутная реализация параллельного ГА в GRID



Менеджер всё ещё должен дожидаться завершения выполнения сторонних программ и получать от них результаты.

Диспетчер GRID, как правило, не предоставляет функций по обмену данными даже между единицами диспетчеризации, не говоря уже о составных частях маршрута.

Подпрограммам генетического алгоритма и внешним средствам САПР необходимо обмениваться следующими данными об эволюционирующих особях: генотип, фенотип, значение фитнес-функции, а также служебной информацией для селективной части ГА (возраст, принадлежность к острову).

Организация обмена данными

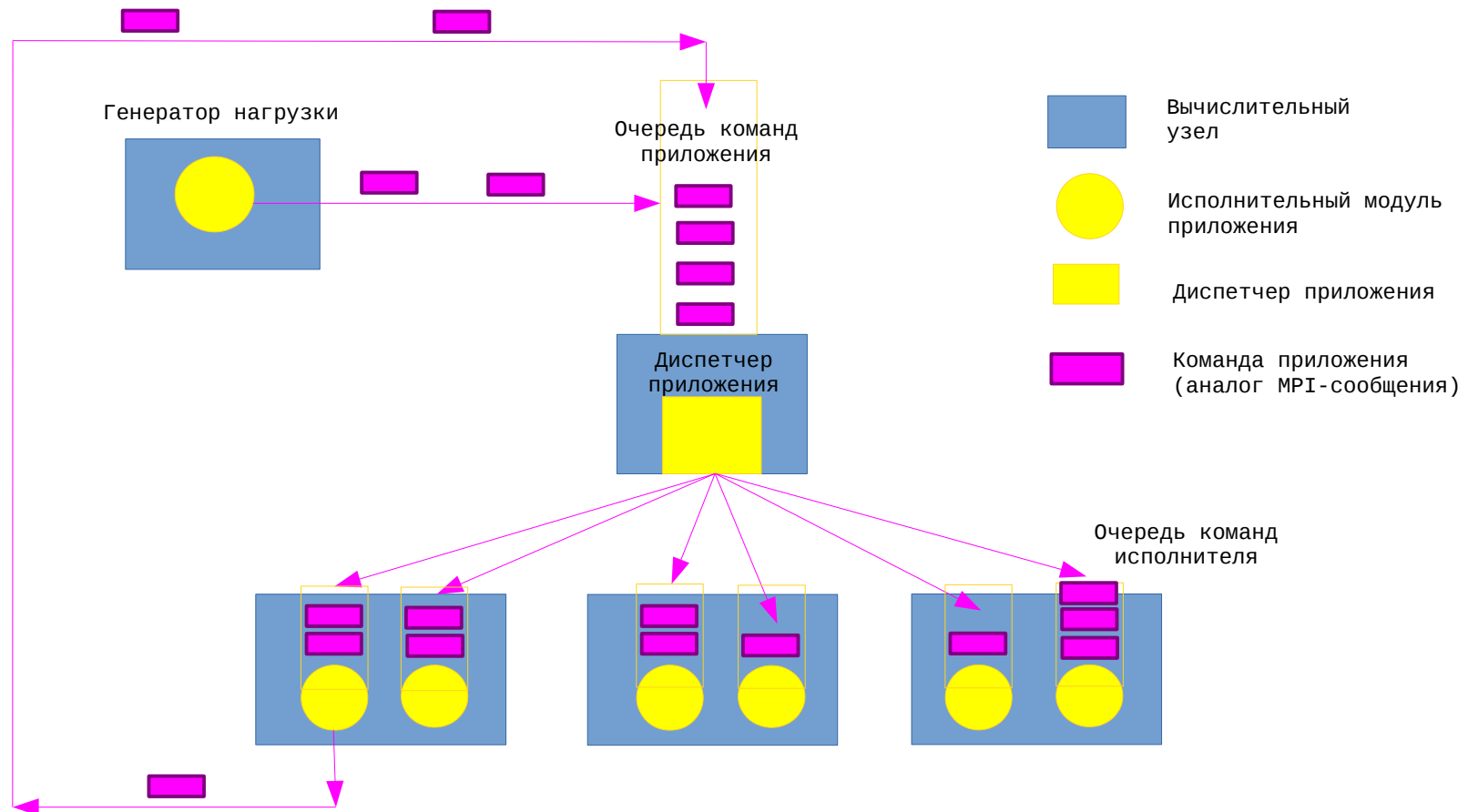
- блокировки в СУБД
- Стандартные механизмы IPC (RPC)
- TSP/IP
- Файлы-флаги

Использование внешних средств САПР заставляет организовывать обмен посредством работы с файлами, находящимися на СХД, общей для всех вычислительных узлов GRID, то есть сетевой файловой системе типа NFS, или, в редких случаях, на кластерной файловой системе типа Ceph, Lustre или GlusterFS.

Реализовать поверх файловой системы стандартные примитивы синхронизации типа семафоров и мьютексов, которые были бы независимы от типа применяемой файловой системы, устойчивы к гонкам, и не вызывали бы серьезную деградацию производительности невозможно, поскольку требование устойчивости к гонкам означает использование синхронных режимов работы файловой системы и отключение кешей на всем пути следования данных, т. е. и на вычислительных узлах и на файл-сервере. Кроме того, выполнение этих условий невозможно проконтролировать при работе в облачной инфраструктуре.

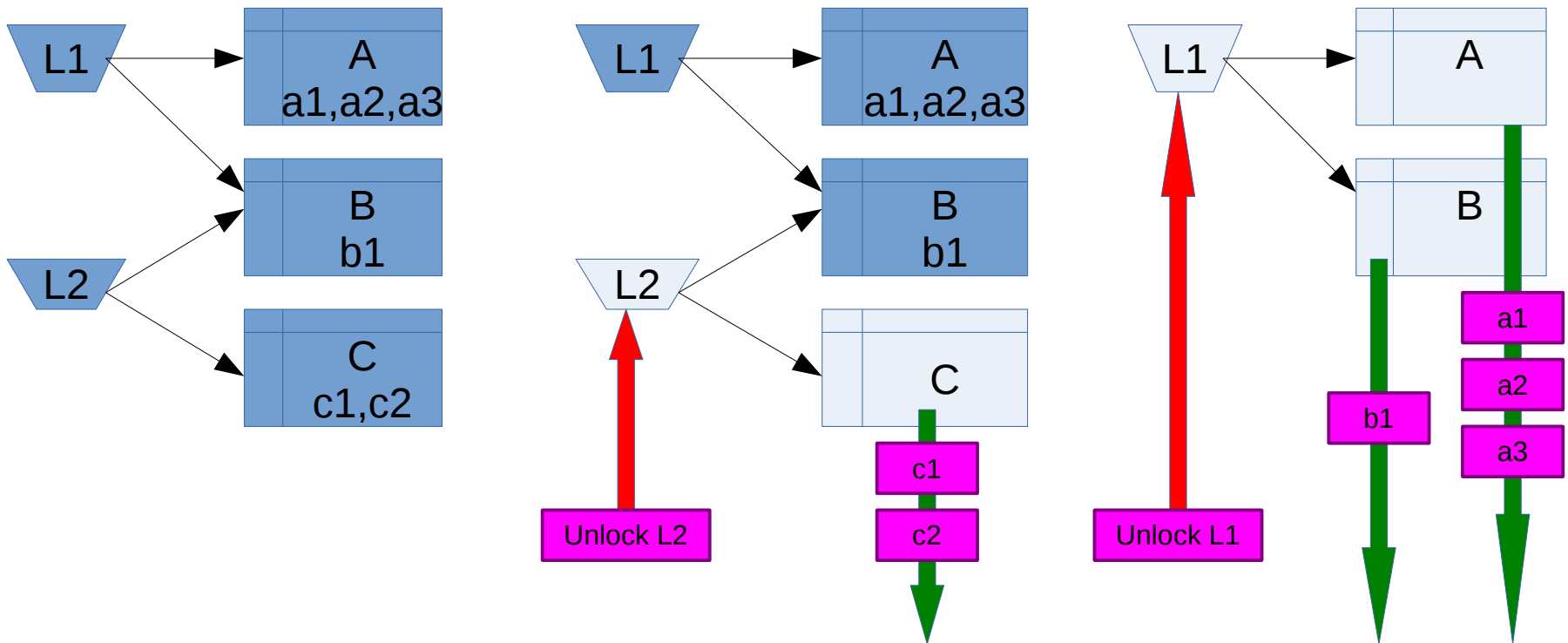
Перед отправкой в GRID задания по обработке особи с идентификатором PersonID создается флаг с помощью команды «ln -s NextGen PersonID» или ее API-аналога. В ходе выполнения маршрута флаг сбрасывается к помощью команды удаления файла «rm PersonID». Чтобы приступить к следующему шагу эволюционного цикла менеджер ожидает завершения обработки всех особей, проверяя условие отсутствия символических ссылок на несуществующий файл NextGen (аналог команды ls -l | grep NextGen). Следует отметить, что в наиболее распространенном случае, когда для обмена данными используется сетевая файловая система NFS, приложение-менеджер, выполняющееся на одном вычислительном узле узнает о том, что файл-флаг был удален маршрутом обработки особи, выполнявшимся на другом вычислительном узле, только после того, как будет обновлен кеш NFS-клиента. В случае традиционной для САПР Linux-среды при настройках по умолчанию это произойдет через 60 секунд. Такие задержки резко ограничивают эффективность работы с множеством небольших задач. Для преодоления этого недостатка может применяться либо тюнинг режимов работы NFS, оказывающий соответствующее негативное влияние на производительность СХД, либо создание отдельного программного средства, отвечающего за блокировки и обмен, которое может быть построено на СУБД, или непосредственно на сетевых протоколах более низкого уровня.

Архитектура эластичного приложения

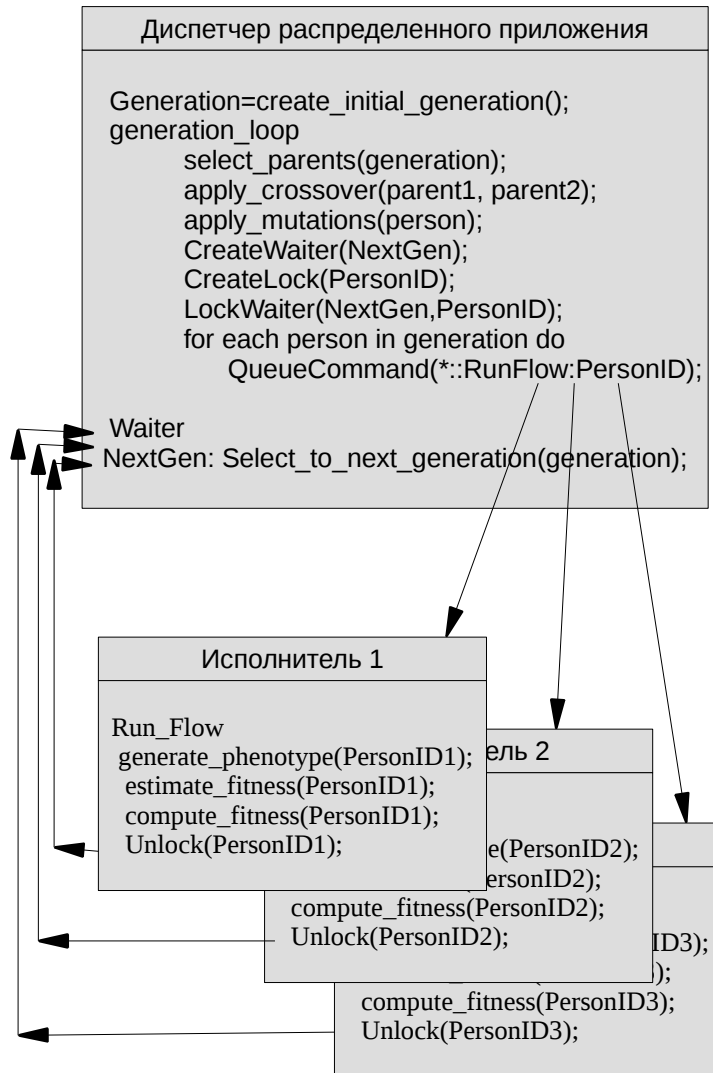


Примитивы синхронизации

- CreateWaiter (Имя официанта, список команд)
- CreateLock (Имя блокировки)
- LockWaiter (Имя официанта, имя блокировки)
- UnLock (Имя блокировки)

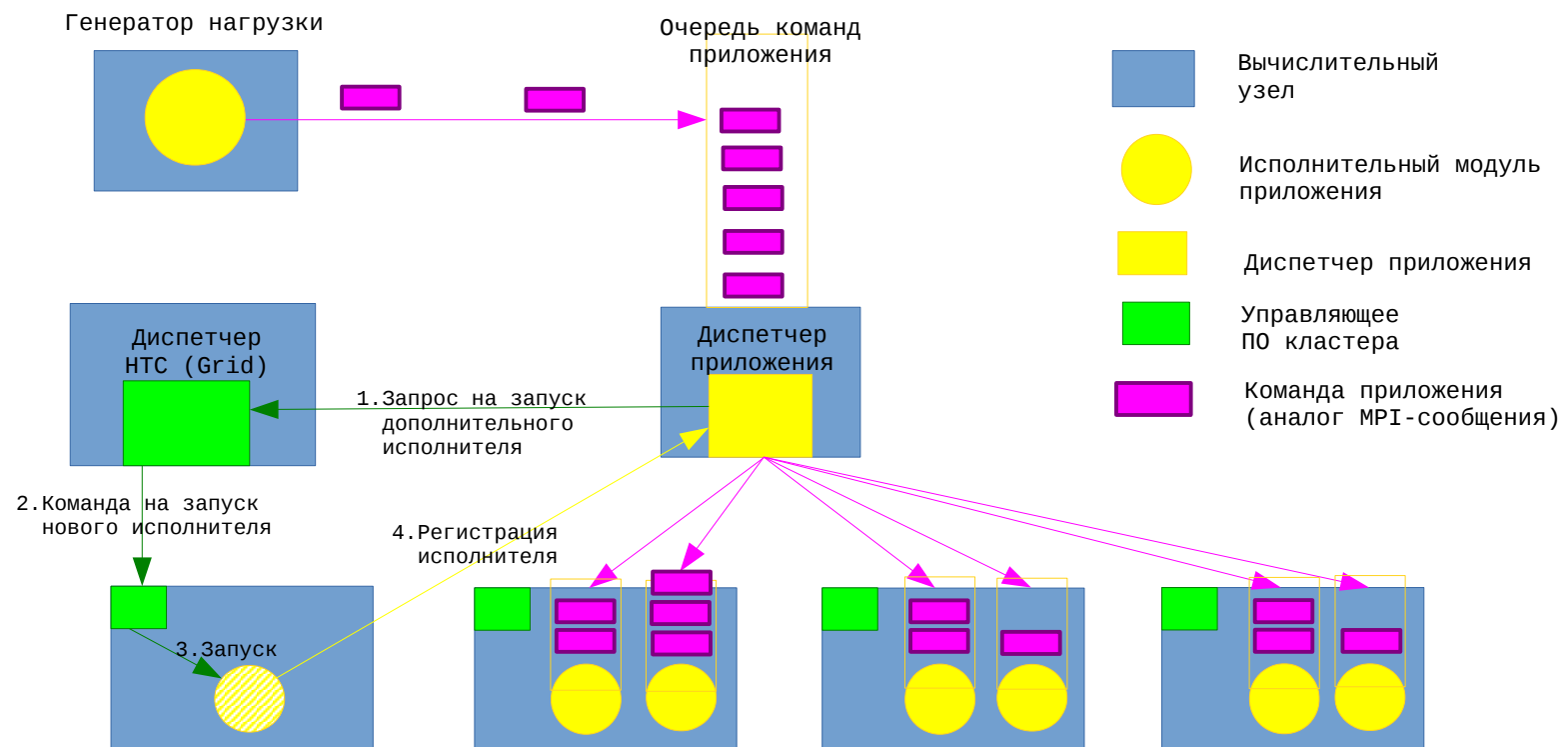


Эластичная реализация ГА



Латентность встроенного механизма обмена сообщениями между исполнителями сильно зависит от загрузки узла, отвечающего за балансировку загрузки. Для информационной инфраструктуры, построенной на базе типовых компонентов (локальная коммутируемая сеть Ethernet 1ГБит/с) время отправки сообщения стабильно ниже 1 мсек, время эхо-ответа же, при тестах варьировалось 10 до 300 мсек. Это на порядки лучше результатов, достигаемых с помощью NFS и сравнимо с производительностью промышленных средств GRID, но далеко от значений, характерных для телекоммуникационных приложений, работающих в такой же инфраструктуре.

Совместное использование архитектуры GRID и эластичного приложения



Эластичное приложение в облаке IaaS

