

Дипломные работы

Темы / направления

1. Основанное на анализе кода fuzzy-планирование потоков с применением `lincheck` для обнаружения ошибок в многопоточном коде
2. Итераторы в lock-free контейнерах (Multi Array): в структуре данных “Feldman's Multi Array” в `libcdds` добавить возможность работы с ключами переменной длины через список коллизий в узлах. Тип lock-free списка должен задаваться извне (в Traits) - сейчас есть три реализации - `MichaelList`, `LazyList`, `IterableList`
3. Пересмотр текущих чистых lock-free алгоритмов в сторону Hardware Transactional Memory
4. Доработки `hpx`
 - Оптимизация планировщика по задачам типа [этой](#)
 - Реализация `lock teleportation` и подобных техник для некоторых структур данных
 - Реализация частей [нового стандарта](#)
 - Перенос алгоритмов из `libcdds` в `hpx` (ведётся активное обсуждение с разработчиками обеих)
 - Адаптивная реализация базовых примитивов на основе HTM с fallback обратно
5. Сравнительный анализ:
 - Производительность STM для Java / Haskell / Closure
 - Производительность HTM в OpenJDK
 - Производительность [HTM](#)
 - RCU примитивы по отношению к `RWLock` для различных структур данных (тут придётся немного пару структур может на `rcu` перевести, но это не сложно)
 - Производительность scalable аллокаторов
6. Оптимизация выделения памяти в `jemalloc` и освобождения при делании сего из разных потоков
7. [В проработке] Доработка применения HTM в OpenJDK ([вводная статья](#) и [от того же автора](#), небольшой [вводный доклад](#) по ТМ в принципе). Сама реализация была сделана в 1.8 по [этой задаче](#).

Архив

1. Доработка алгоритмов [Google Thread Sanitizer](#), в частности:
 - В области уборки некорректных срабатываний в lock-free
 - В целом исправление работы с fine-grained-lock алгоритмами
2. Реализация и моделирование сетевых протоколов в симуляторе [NS-3](#)
3. Доработка визуализатора сетевой симулации [NetAnim](#)
4. Доработка алгоритмов детекции Data race в valgrind [helgrind](#) и [drd](#) по существующим ошибкам / запросам
5. Алгоритмы, требующие реализации и доработки в `libcdds`, обычно есть что улучшить и ускорить по сравнению с реализацией из статей

Текущие

Магистранты

Бакалавры

Защищённые

Магистры

1. Ёров С. "Доработка алгоритмов *Google Thread Sanitizer*", АУ 2018
2. Доронин О. "Автоматическое *fuzzy*-планирование потоков с помощью *relacy* для обнаружения ошибок в многопоточном коде", АУ 2018
3. Карулин Н. "Исследование и разработка методов обеспечения заданной производительности системы анализа последовательностей генома", ЛЭТИ 2017
4. Яцык А. "Разработка протокола формирования и передачи вектора состояния ОЭС ТИК-М.", ИТМО 2017
5. Галимуллин М. "Разработка адаптивной стратегии синхронизации потоков в конкурентных структурах данных, основанных на *flat-combining*", ЛЭТИ 2016
6. Рапоткин Н. "Разработка стратегий *flat-combining* для конкурентных структур данных на примере библиотеки *libcds*», ЛЭТИ 2015 (+ Балтрашевич)
7. Леснова О. "Разработка методов балансировки нагрузки для платформы моделирования сетей *NS-3*", ЛЭТИ 2013 (+ Балтрашевич)
8. Алексеева А. "Разработка алгоритма маршрутизации беспроводных *Mesh*-сетей в условиях ограничения на энергопотребление узлов", ЛЭТИ 2013 (+ Балтрашевич)
9. Александрова С. "Разработка средства моделирования пространственной структуры белковых молекул", АУ 2010

Бакалавры

1. Шохин Е. "Интерактивная карта университета с отображением расписания", ЛЭТИ 2016
2. Королёв Ю. "Разработка подсистемы визуализации созвездия навигационных космических аппаратов", ЛЭТИ 2009 (+ Кафтасьев)

From:

<http://wiki.osll.ru/> - Open Source & Linux Lab



Permanent link:

<http://wiki.osll.ru/doku.php/etc:users:kel:diplomas?rev=1584305050>

Last update: **2020/03/15 23:44**