Разработка подсистемы управления объектами для системы моделирования алгоритмов и структур с мелкозернистым параллелизмом

Сафин А. Р.

Научный руководитель
Остапкевич М. Б.

- Система моделирования
- Модели мелкозернистая структуры
  - Клеточные автоматы
  - Микропроцессорные архитектуры
  - Нейросети
- Большие данные

Обзор

- Распределённые операционные системы
  - Mosix
- Координационные языки
  - Linda

Linda

- Распределённая координационная система
- Пространство кортежей
- Операции
  - rd
  - in
  - out
  - eval

```
rd("square", 16, ?a)
out("square", 16, 4)
eval("square", 16, square(16))
eval("worker", worker())
```

MOSIX

- Распределённая операционная система
- Основана на Linux
- Single-system image
- Работает на
  - кластерах
  - мультикластерах
  - облаках
- Миграция процессов

Требования

- адекватность проблемной области
- открытая архитектура
  - расширяемость
  - переносимость
  - интероперабельность
  - дружественность интерфейса
- обработка больших объёмов данных

Менеджер объектов

- Отображает объекты в виртуальную память
- Структуры данных
  - Двумерные и трехмерные массивы
  - Октодеревья
  - …
- Большие объекты
- Хранимый во внешней памяти объект
  - Имеет имя
  - Дескриптор объекта
    - Массив 128K x 128K из float, разбитый на квадраты 4K x 4K
  - Части объекта

Операции менеджера объектов

- Создание и удаление объектов
- Загрузка и сохранение дескрипторов объектов по их имени
- Загрузка и сохранение частей объектов
- Блокировки частей объектов
  - На чтение
  - На чтение и запись

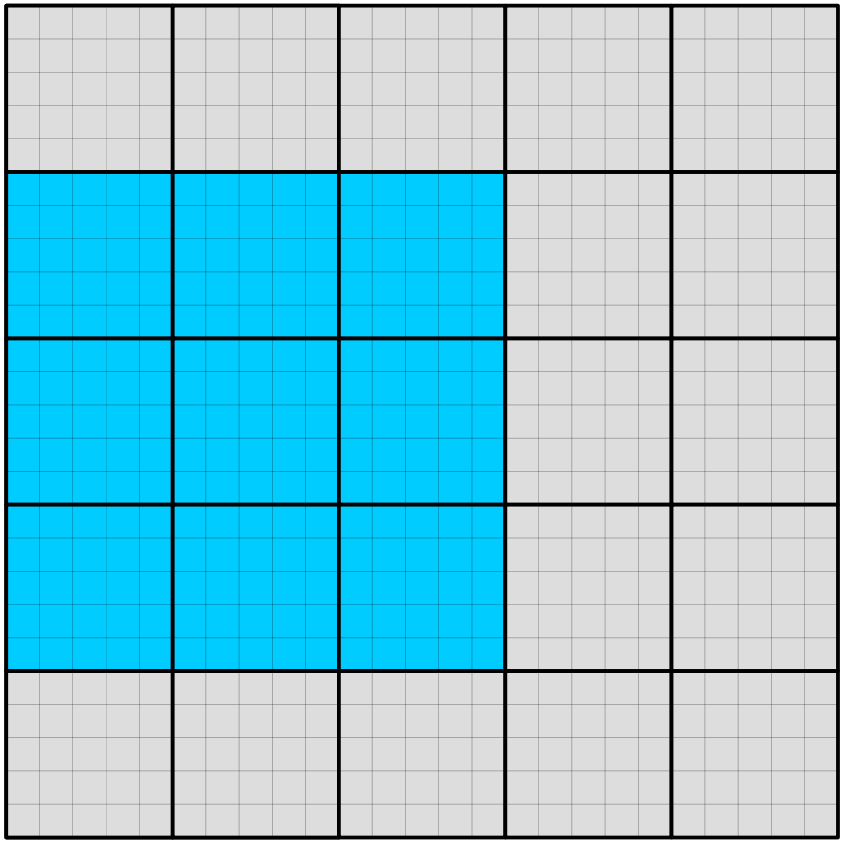
Игра «жизнь»

```
 1 int main(){
 2         ObjectManager om;
 3         Array2D* f0 = om.load_obj("life");
 4         Array2D* f1 = om.load_obj("life_");
 5
 6         for(int i = 0; i < 100; i++) {
 7                 life_step(f0, f1);
 8                 life_step(f1, f0);
 9         }
10
11         om.unuse_obj(f0);
12         om.unuse_obj(f1);
13 }
```

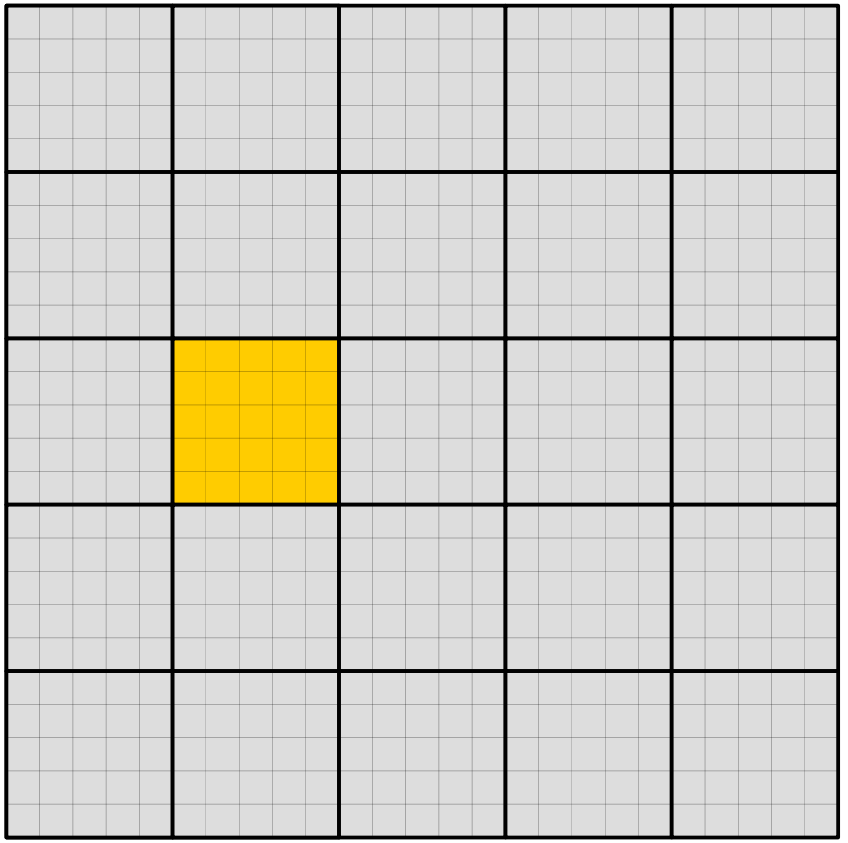
Игра «жизнь»

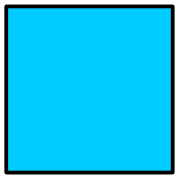
void life_step(Array2D *in, Array2D *out)

in

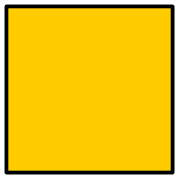


out





R lock

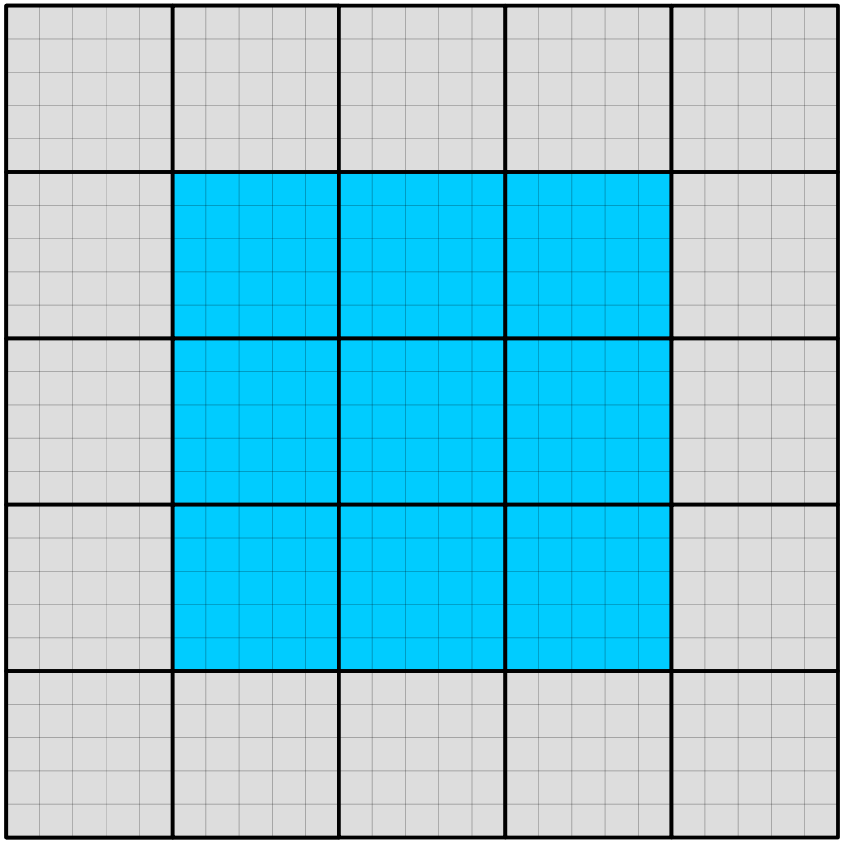


R/W lock

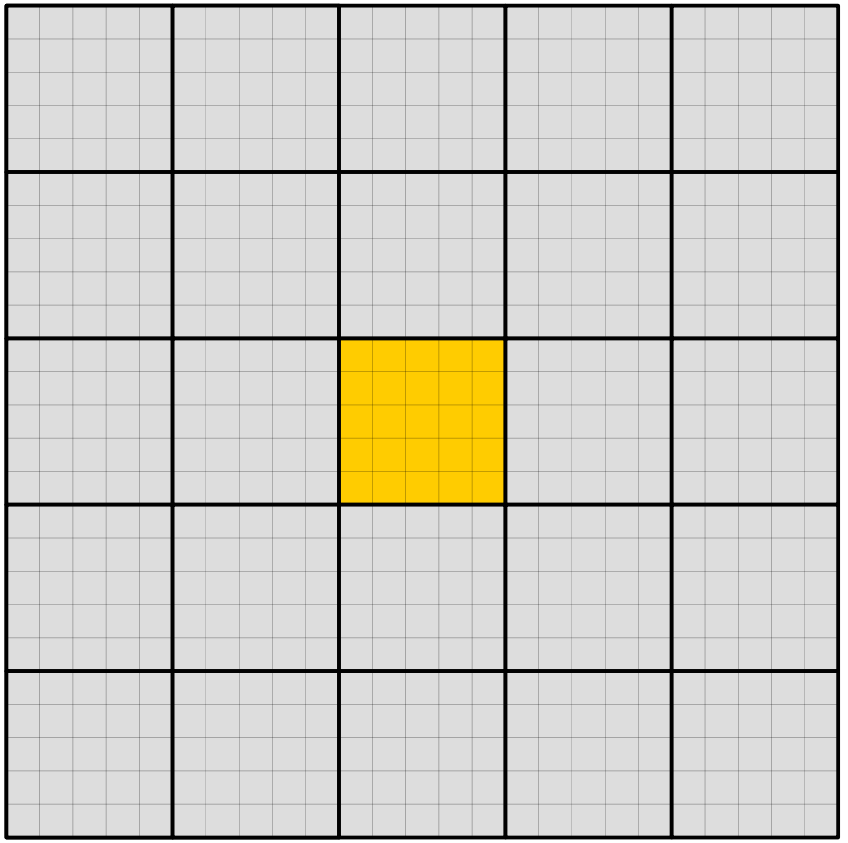
Игра «жизнь»

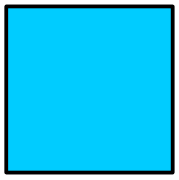
void life_step(Array2D *in, Array2D *out)

in

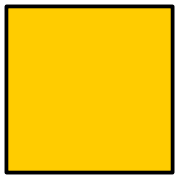


out





R lock



R/W lock

Игра «жизнь»

```
1  void life_step(Array2D *in, Array2D *out) {
2    for(/* x,y ... */) {
3      for(/* i,j ... */) in->lock_r_block({x+i, y+j});
4      out->lock_rw_block({x, y});
5
6      const char* in_[9] =
7          { in->blocks_data[{x-1, y-1}],
8            /* ... */,
9            in->blocks_data[{x+1, y+1}] };
10     char* out_ = out->blocks_data[{x, y}];
11     small_life_step(in_, out_);
12
13     for(/* i,j ... */) in->unlock_r_block({x+i, y+j});
14     out->unlock_rw_block({x,y});
15   }
16 }
```

- Вывод
  - Сделан обзор
  - Описан и реализован интерфейс локального менеджера объектов
  - Тестовый пример «Жизнь»
- Планы:
  - Распределённый менеджер объектов
  - Более серьёзная модель для тестов