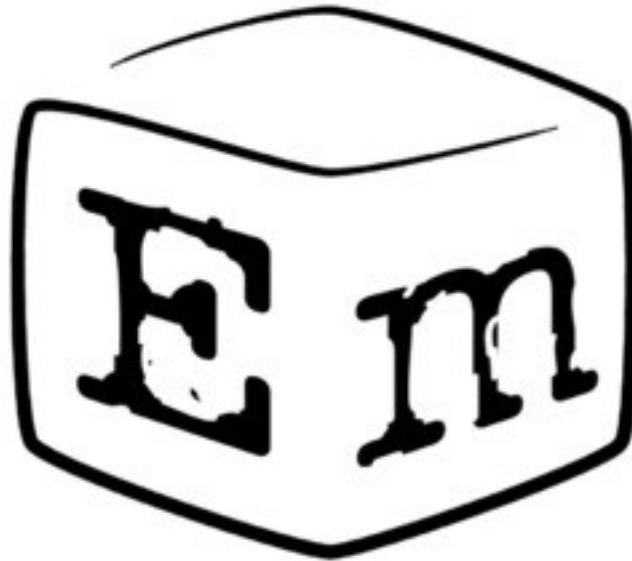


Курсовая работа

на тему: "Оптимизация вычислений за счет эффективных структур данных в ОСРВ Embox"



Мальковский Н.В., 445гр.
научный руководитель: Бондарев А.В.

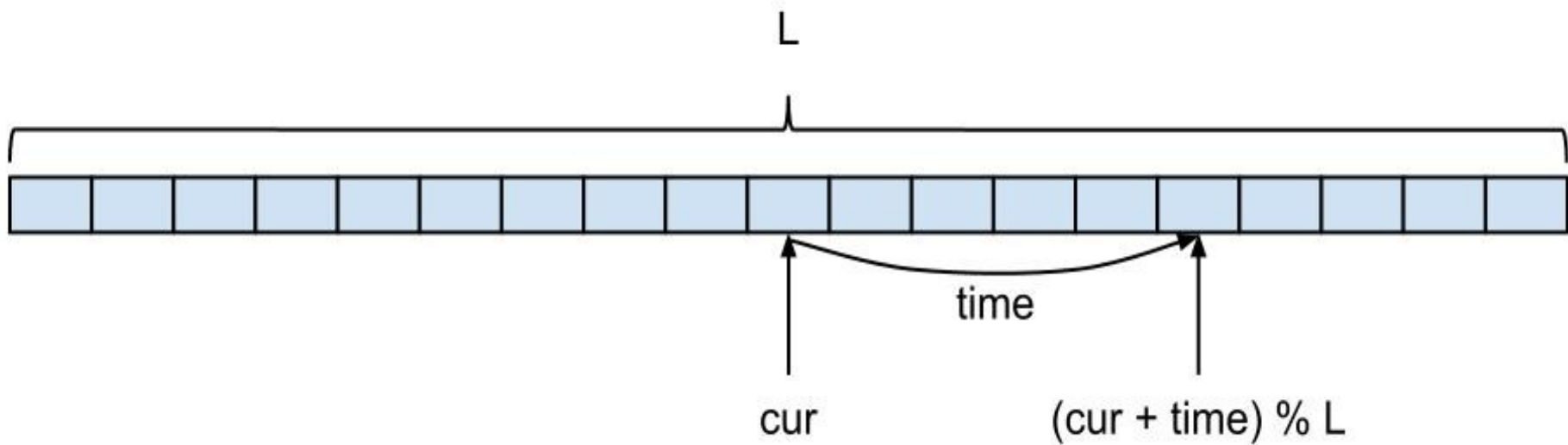
Планировщик событий - это структура, поддерживающая 2 операции:

- **Tick()** - Счётчик, является источником времени для структуры - для нее время измеряется в количестве вызовов этой функции
- **Insert(time, handler)** - добавить в структуру событие, которое должно быть обработано через **time** вызовов **Tick** с помощью функции **handler**
- ***Delete(handler)** - часто нужно уметь удалять уже добавленные события

Планировщик в Embox

- Отсортированный в хронологическом порядке список всех событий
- Для всех событий, кроме первого, храним разницу моментов времени обработки с предыдущим событием
- Декрементируем только голову списка
- При добавлении нужно пройти по списку и найти место, куда следует поместить новое событие

Простой алгоритм



Tick: $cur = (cur + 1) \% L$, $dispatch(cur)$;

Insert($time$, $handler$): $list_add(arr[(cur + time) \% L], handler)$;

Свойства алгоритма

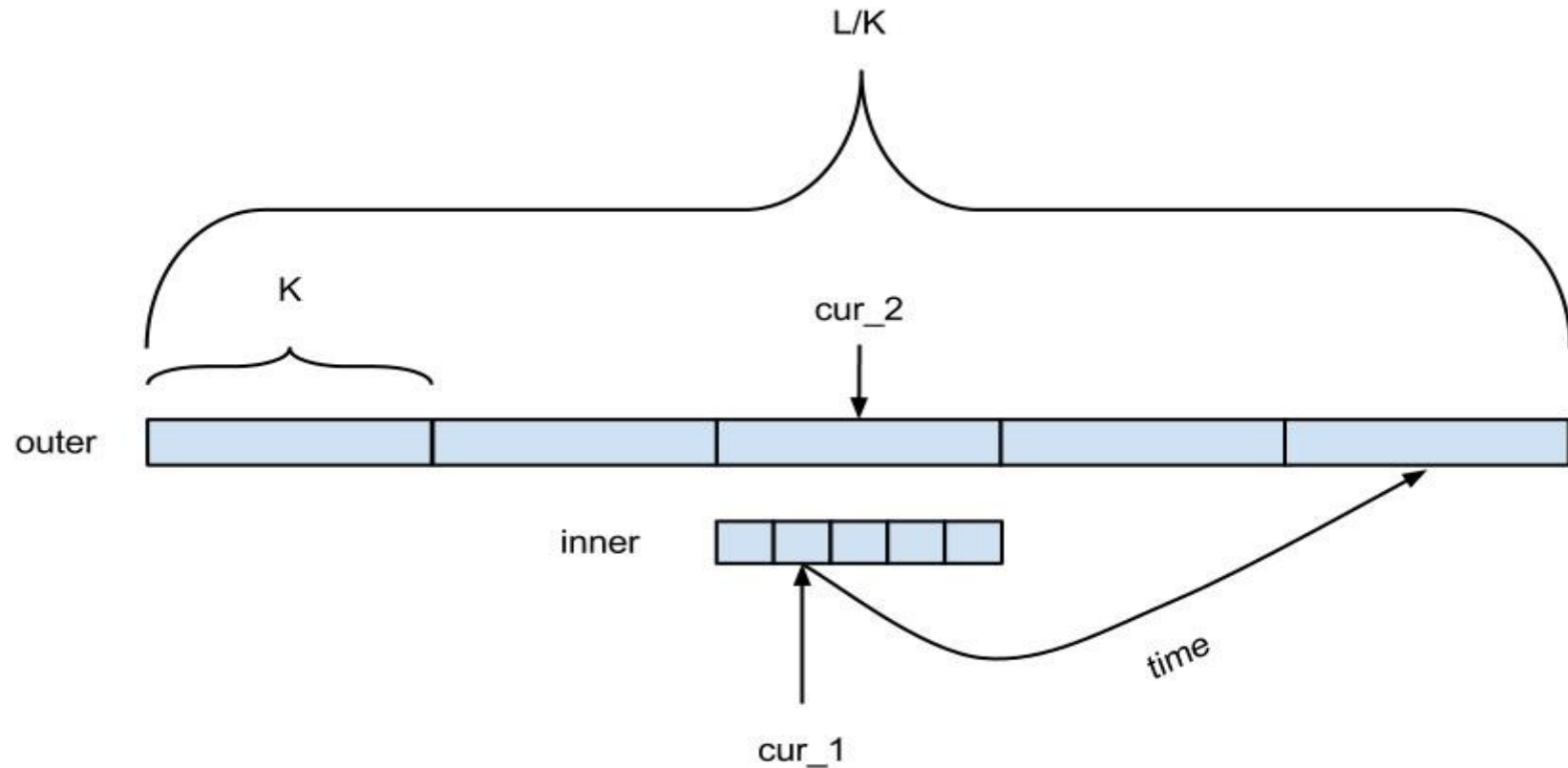
- **Плюсы**

- Простота реализации
- Минимальные затраты процессорного времени

- **Минусы**

- Относительно большие расходы памяти

Основной алгоритм



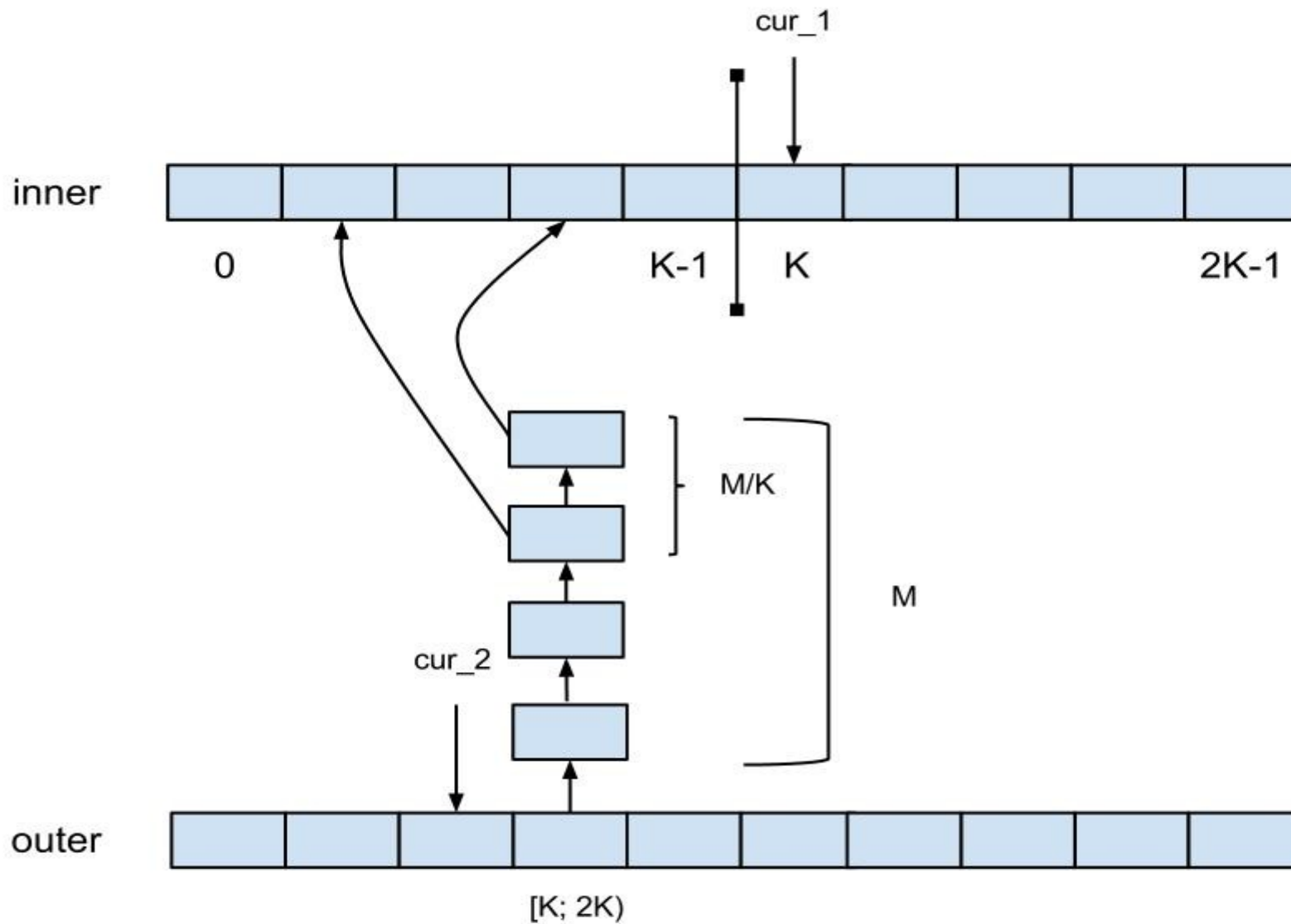
Insert(time, handler):

```
if(time < K) list_add(inner[(cur_1+time)%K],handler);  
else list_add(outer[(cur_2+(cur_1+time)/K)%(L/K)], (cur_1+time)%K, handler);
```

Tick:

```
cur_1 = (cur_1 + 1) % K;  
if(cur_1 == 0) cur_2 = (cur_2 + 1) % (L / K), transfer(outer[cur_2]);
```

Амортизация



Результаты

- Эффективный и предсказуемый алгоритм планирования событий
 - На обработку одного события всегда приходится 1-2 добавление(удаление) в список(из списка) + несколько арифметических операций
 - Обработка N событий за K вызовов Tick осуществляется за $O(K+N)$, что по сути есть размер входных данных
- Можно рассматривать как приоритетную очередь