

RAST: highly distributed DB proejct.

Produced by NOA team

Contents

1	Algorithmic basis	2
1.1	Allocator	2
2	Detailed Architecture (classes and methods)	2
2.1	BulkAllocator	2
2.2	Lock-free queue	3

1. Algorithmic basis

1.1. Allocator

2. Detailed Architecture (classes and methods)

2.1. BulkAllocator

123123 BulkAllocator is an allocator, which for each required type size stores a list of

2.2. Lock-free queue

LockFreeQueue < *TElement* > is obviously an implementation of lock-free queue. It contains the following functions:

```
LockFreeQueue<TElement> {  
    void Push(TElement new_element);  
    // Exceptions:  
    //     Strong exception safety: in case of failure  
    //     the element is not pushed to the queue,  
    //     there are no visible side effects  
  
    std::unique_ptr<TElement> Pop() noexcept;  
    // Return value:  
    //     nullptr unique_ptr if the queue was empty,  
    //     unique_ptr pointing on TElement in case of successfull pop  
};
```
