Allocators

TODO

Henruch Lauko, Jiří Novotný, Katarína Kejstová

May 19, 2016

Motivation

-
-
-

C++ Allocators API

- allocated memory representation:

  ```
  struct Block { void* ptr; size_t size; }
  ```

- requirements on allocator:

  ```
  Block allocate(size_t)
  void deallocate(Block)
  bool owns(Block)
  ```

Allocators

```
NullAllocator

Mallocator

StackAllocator<size_t size, size_t alignment>

FallbackAllocator<class Primary, class Fallback>

Freelist<class Allocator, size_t min,
         size_t max, size_t capacity>
```

Allocators

```
Segregator<size_t threshold,
           class SmallAllocator, class LargeAllocator>

AffixAllocator<class Allocator,
               typename Prefix, typename Suffix>

StatisticCollector<class Allocator, int Option>

BitmappedBlock<class Allocator, size_t block_size>
```

Freelist problems

- TODO problems

BitmappedBlock

Modularity – composability

- composition of allocators, specialized by block sizes
- arrays, lists, trees of allocators

```
using Allocator =
        Segregator<1024,
            Segregator<512,
                Freelist<Mallocator, 0, 512,
                Freelist<Mallocator, 513, 1024>
            >,
            Mallocator
        >;
```

Example

Benchmarks