Memory Management of Fools Engine

Work in progress

Areas with custom memory allocation strategies

Area		Allocation strategy
Gameplay World	Components	ECS A
	Behaviors, Systems and Components' dynamic data	"Isolator" allocator A
Assets	Components	ECS B
	Components' dynamic data	"Isolator" allocator B
	Assets' Data	Virtual multipool allocator
Frame lifetime related memory	Full pipeline	Ring buffer allocator
	Single stage	Monotonic allocator
Very short-lived memory		"Scratchpad" allocator

ECS

EnTT library with sparse set data structure per component type with manual memory pagination directly from malloc, and that is OK in this case.

Frame pipeline single stage related lifetime memory – Monotonic allocator

std::pmr::monotonic_buffer_resource with static initial buffer and malloc as upstream allocator. Reset after every stage with release of additional buffers.

Very short-lived memory
Scratchpad allocator

Basic idea:

Thread local stack of bump allocators
compatible with std::pmr

```
{
    Scratchpad sp_1;
    void* p1_1 = sp_1.allocate(2);
    void* p1_2 = sp_1.allocate(3);
    void* p1_3 = sp_1.allocate(1);
    void* p1_4 = sp_1.allocate(4);
    {
        Scratchpad sp_2;
        void* p2_1 = sp_2.allocate(2);
        void* p2_2 = sp_2.allocate(3);
        void* p2_3 = sp_2.allocate(4);
        void* p2_4 = sp_2.allocate(1);
    }
}
```

Very short-lived memory
Scratchpad allocator

Obvious issue:

Interlocking allocations

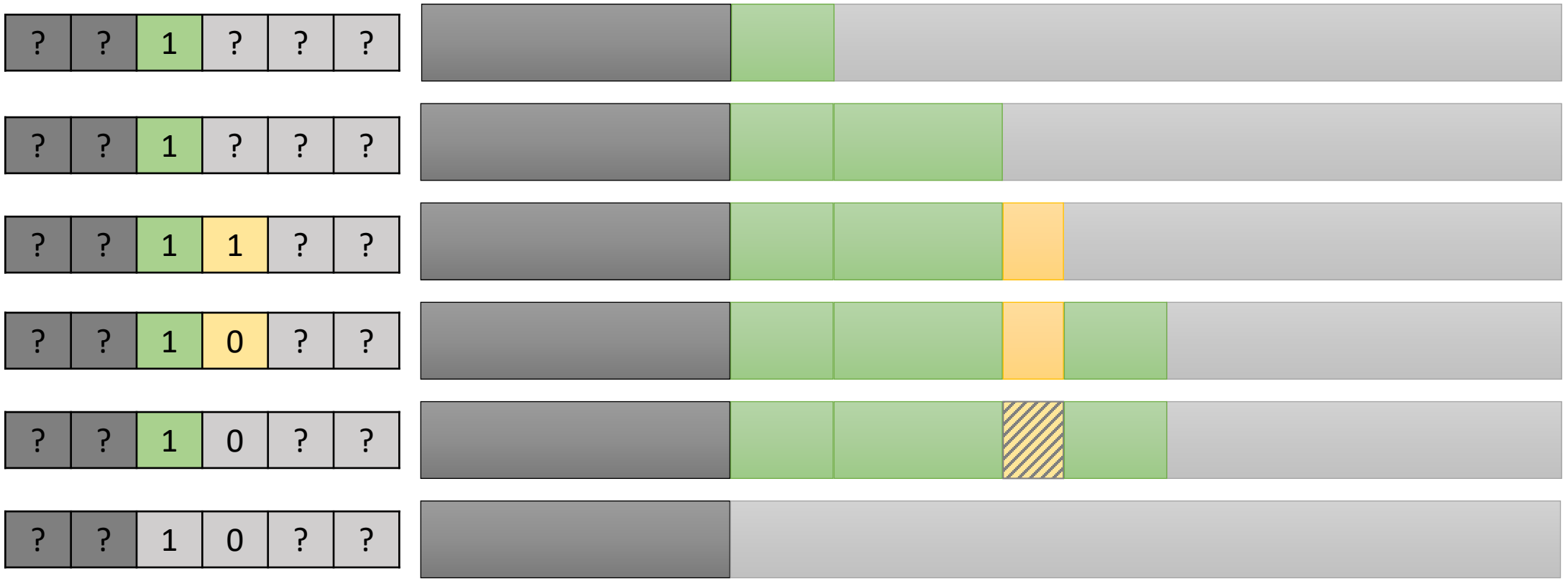
```
Scratchpad sp_1;
Scratchpad sp_2;
void* p1_1 = sp_1.allocate(2);
void* p2_1 = sp_2.allocate(2);
void* p1_2 = sp_1.allocate(3);
void* p2_2 = sp_2.allocate(3);
void* p1_3 = sp_1.allocate(1);
void* p2_3 = sp_2.allocate(4);
```

Can happen implicitly when Scratchpad allocator is passed to std::pmr containers (allocation when they grow)!

Very short-lived memory
Scratchpad allocator

Solution:

- Bitflag per allocator indicating whether to roll back the pointer or not
- Every allocation sets flags of younger allocators to false

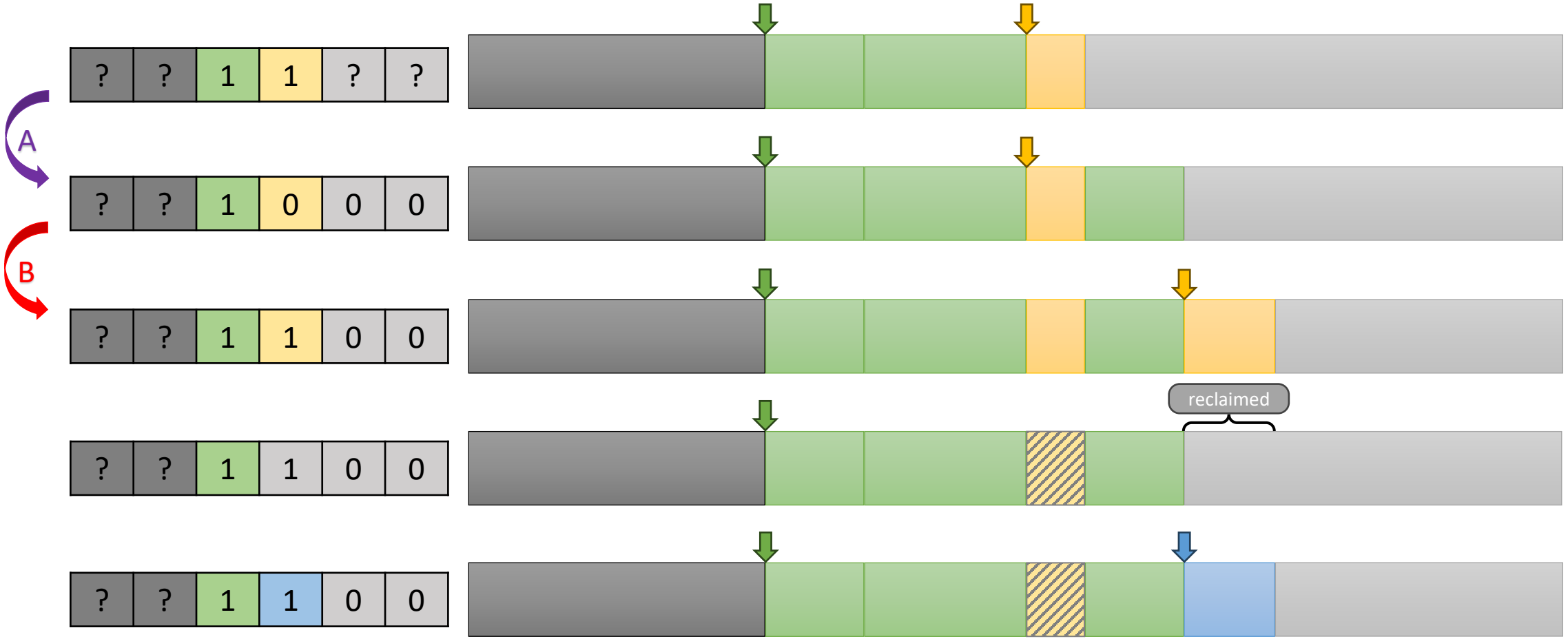


Very short-lived memory
Scratchpad allocator

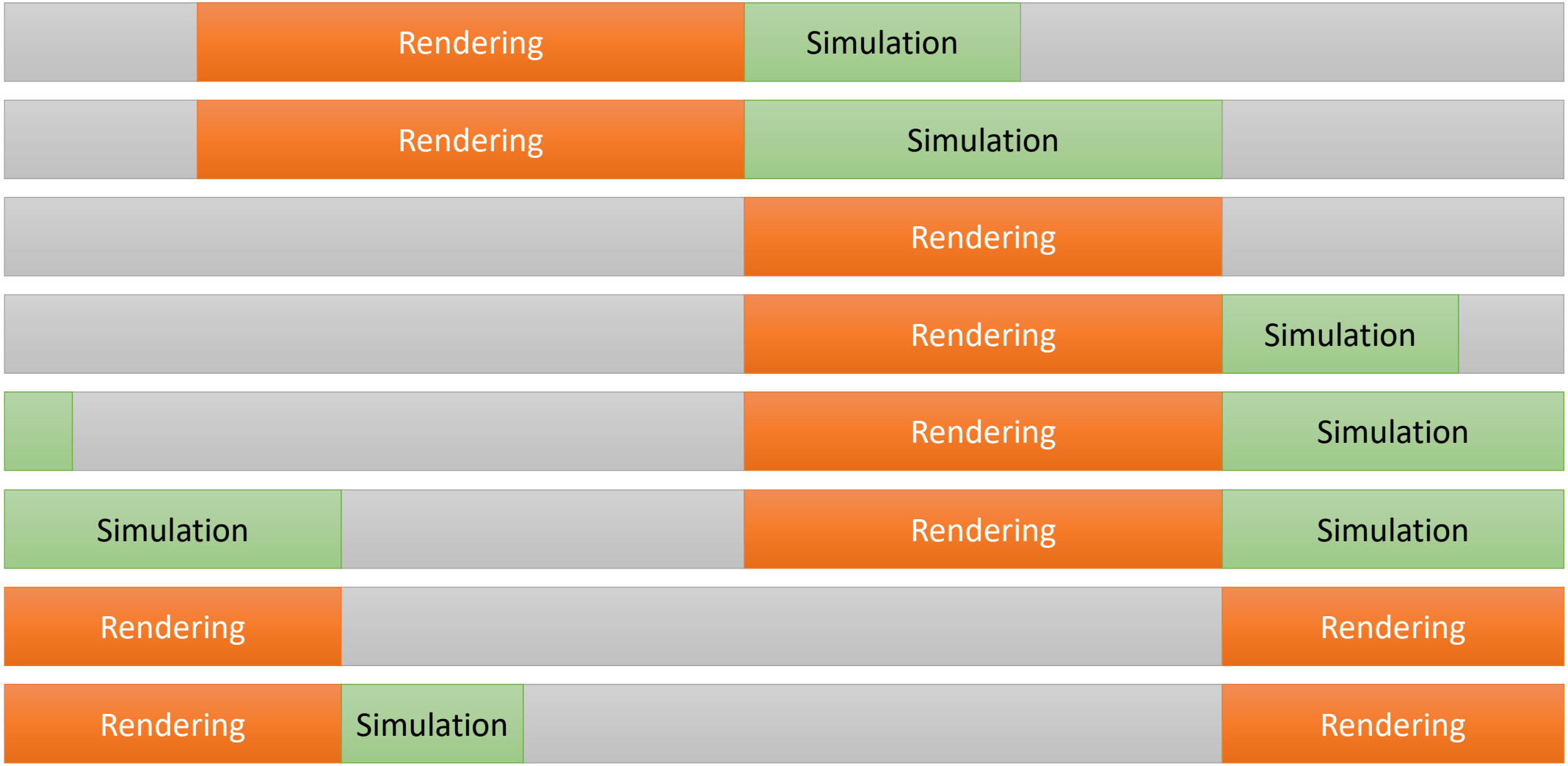
Optimalizations:

A. Setting rollback flags to false on all positions to the right regardless of allocators' existence

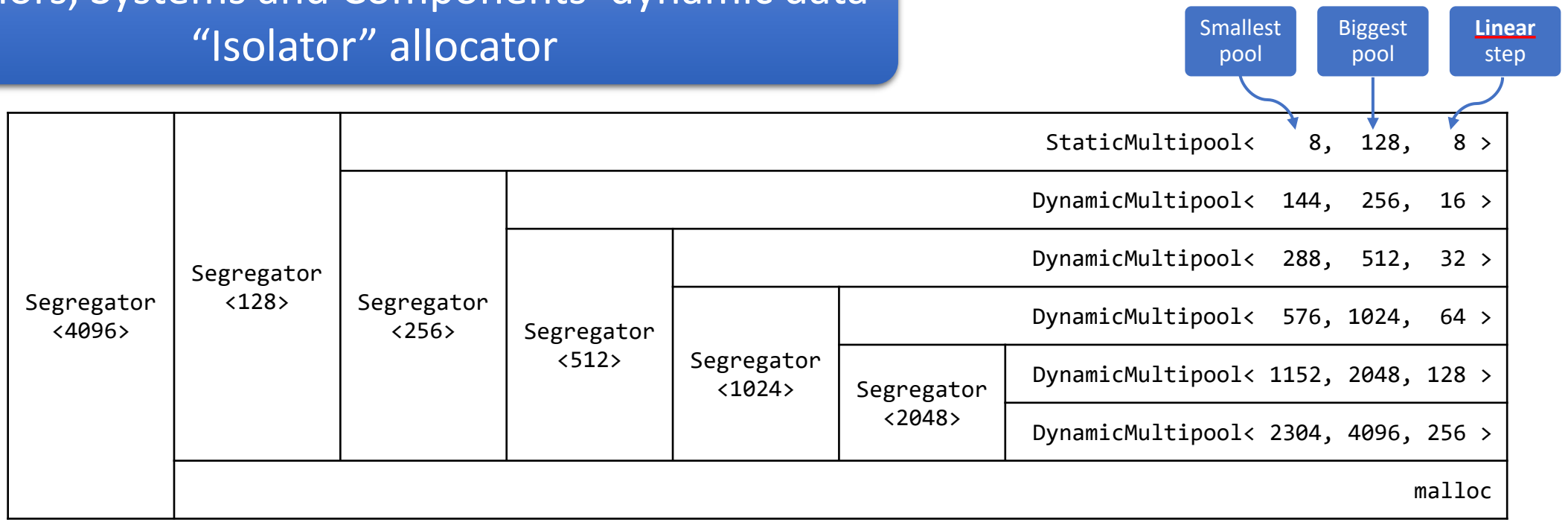
B. Resetting rollback flag to true and rollback pointer when "reclaiming" front



Frame pipeline related lifetime memory
Ring buffer allocator



Behaviors, Systems and Components' dynamic data "Isolator" allocator



StaticMultipool

- Fixed array of pools
- Pools keep chunks forever

DynamicMultipool

- Fixed array of pointers to pools initialized with nullptrs
- Chunks keep track of allocated blocks count and get released when empty
- When whole pool is empty, it gets destroyed and released too