facebook

G1 Tuning and Real-Time Services

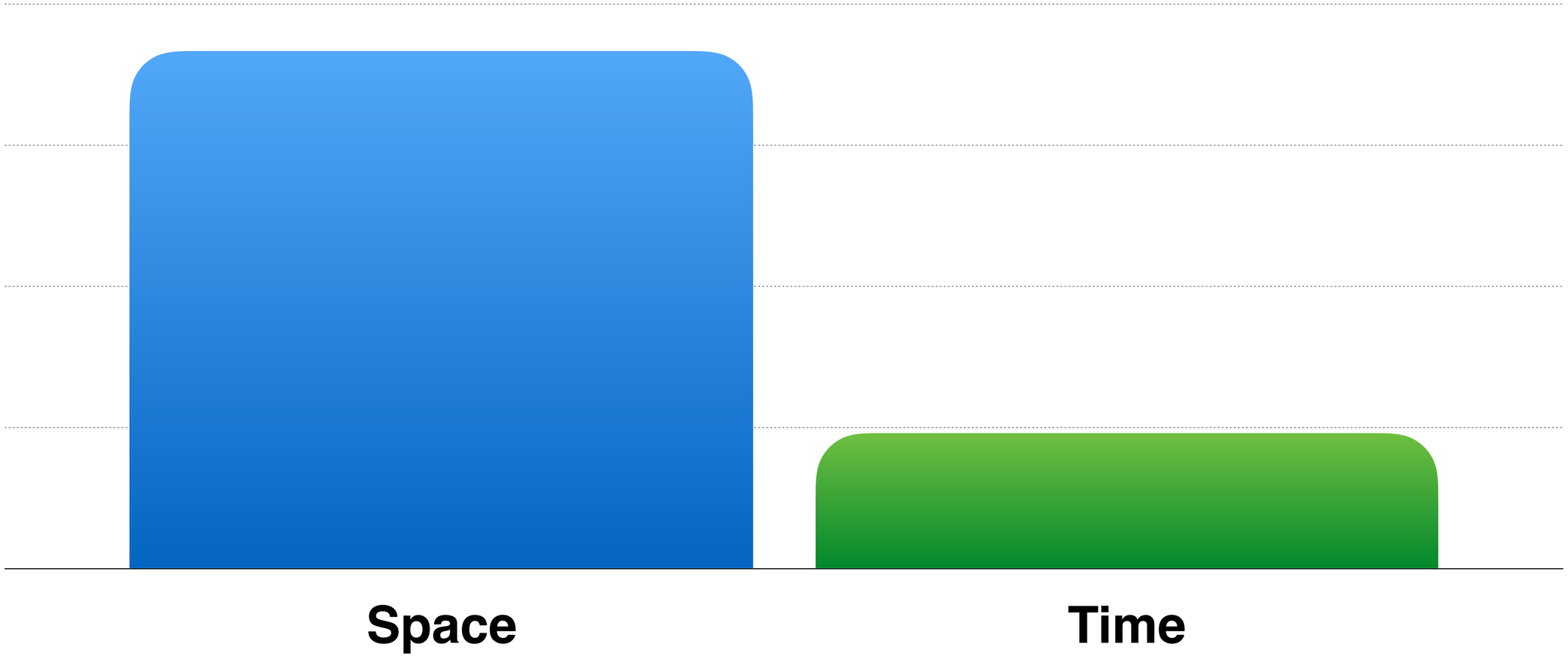
Andi Chalfant

Production Engineer, Messaging Infrastructure

Scaling Means...

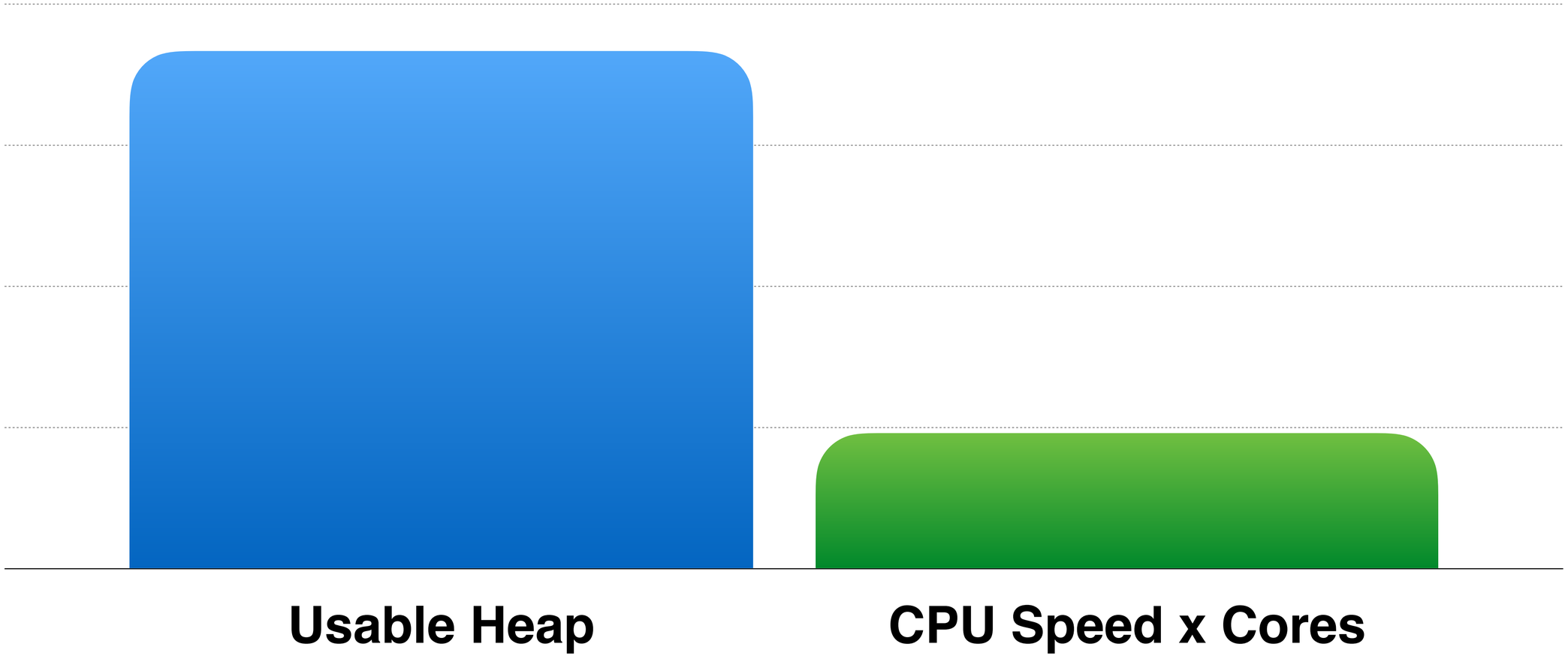
What trade-offs will you make?

Space vs Time!



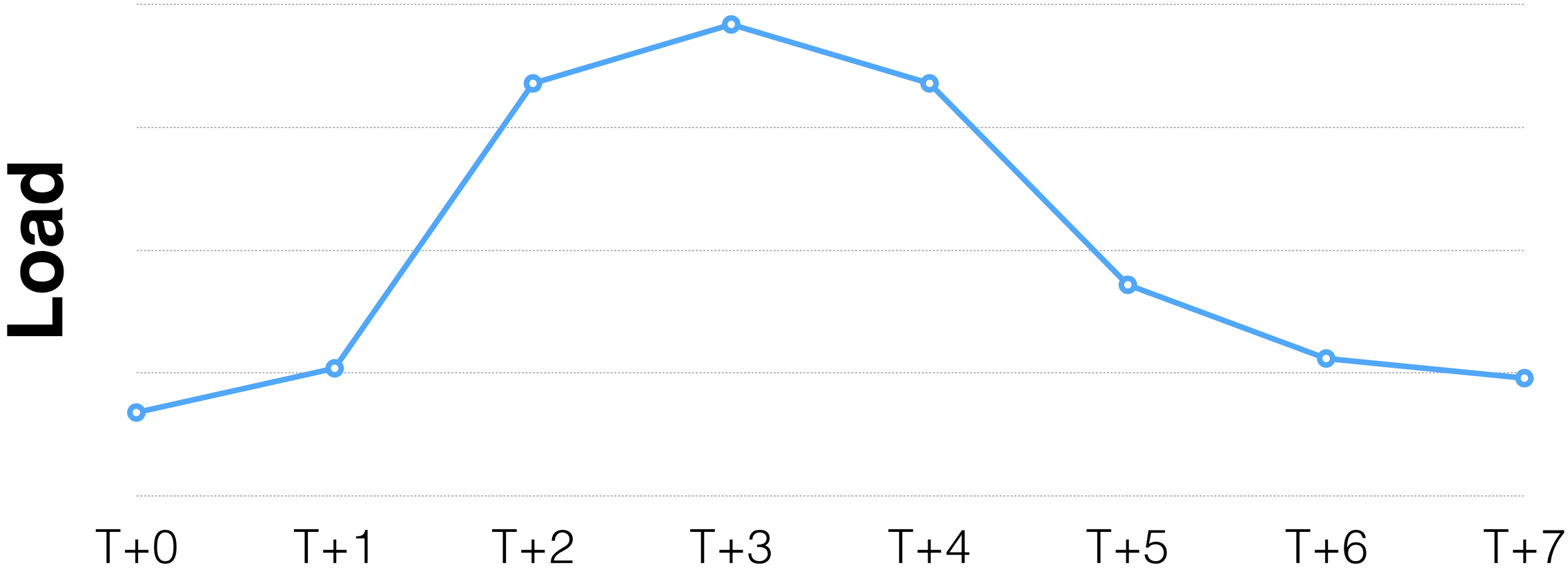
What trade-offs will you make?

Space vs Time!



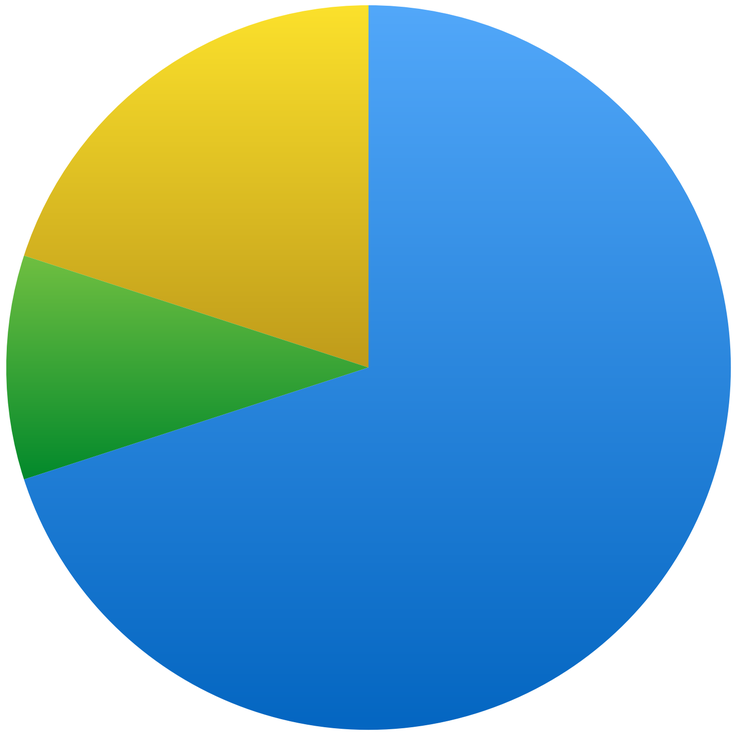
What's your load profile?

Load varies by time!



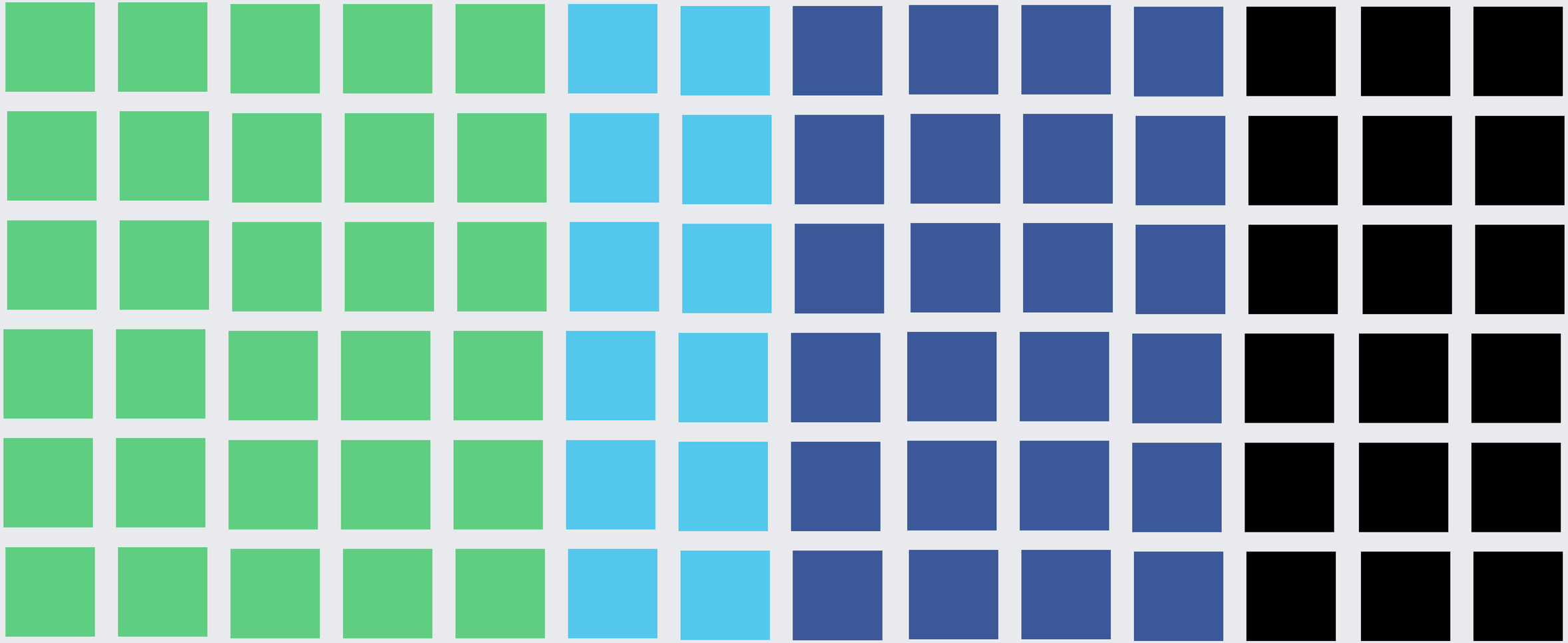
What's the minimum your tax needs?

- Your Service
- Page Cache
- Bloat-with-Features!

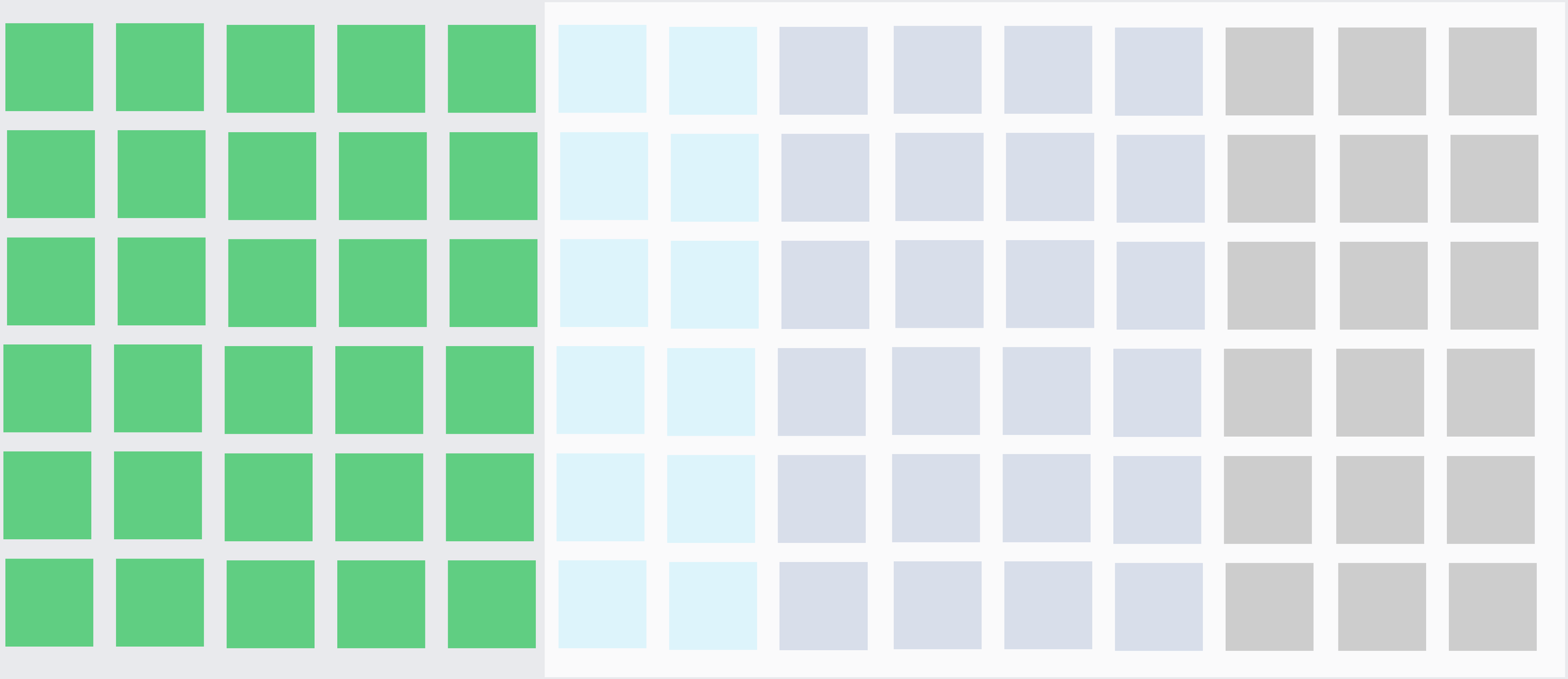


G1: The Future Of Yesterday, Today!

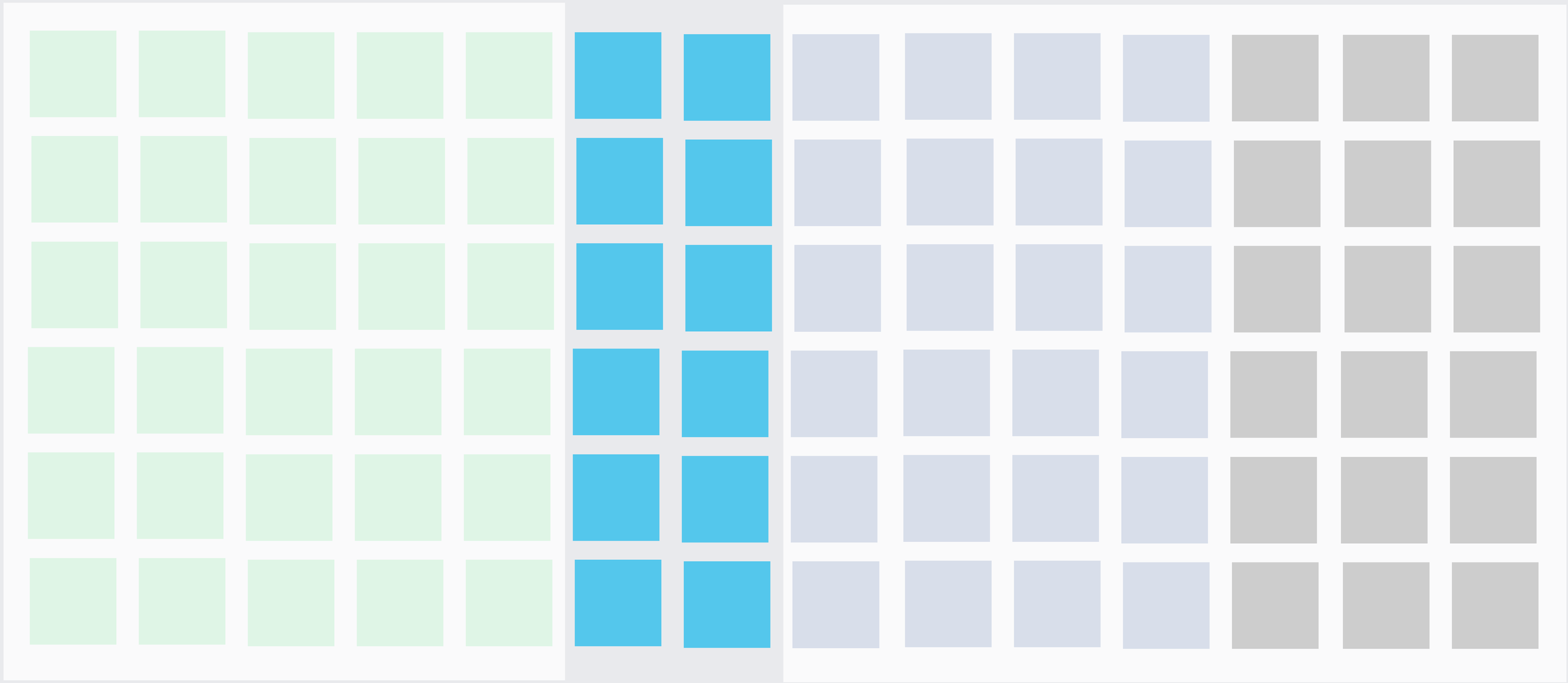
The G1 Heap Layout: Regions, Generations



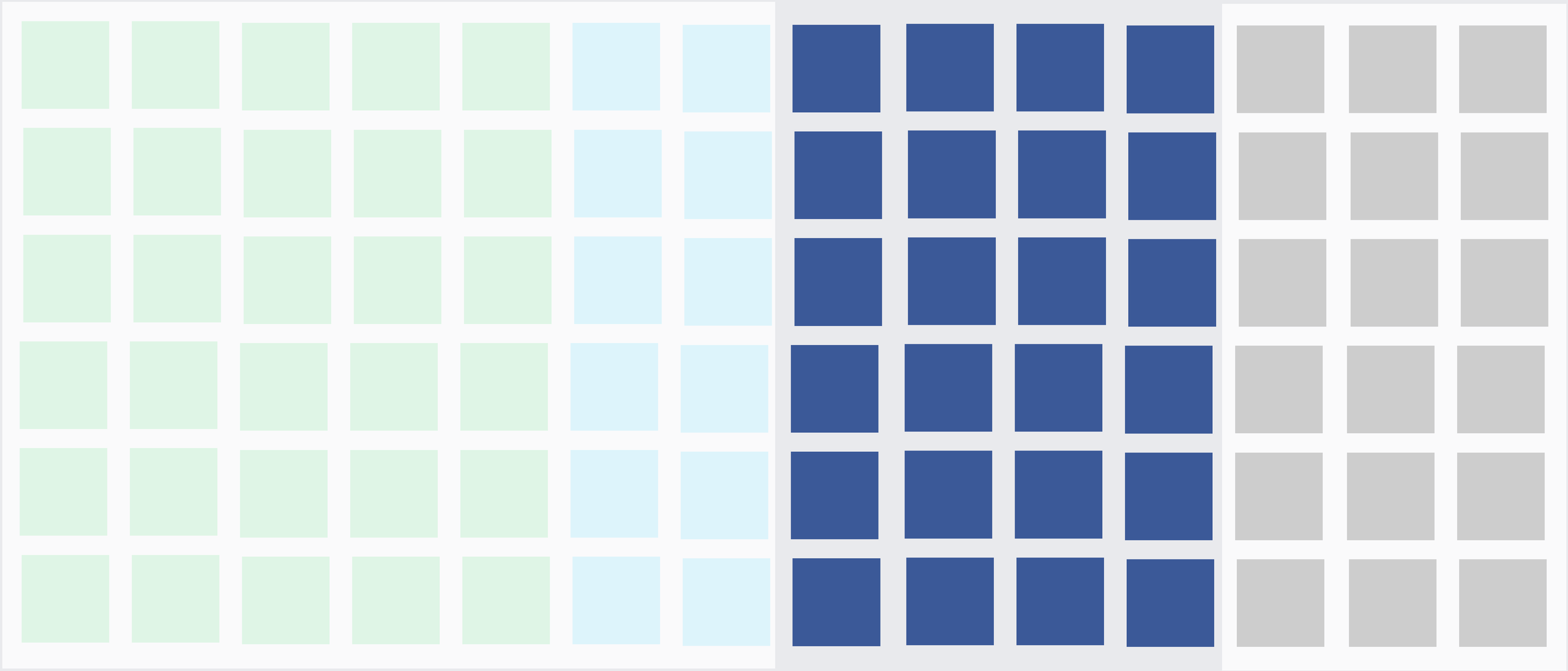
Eden Space



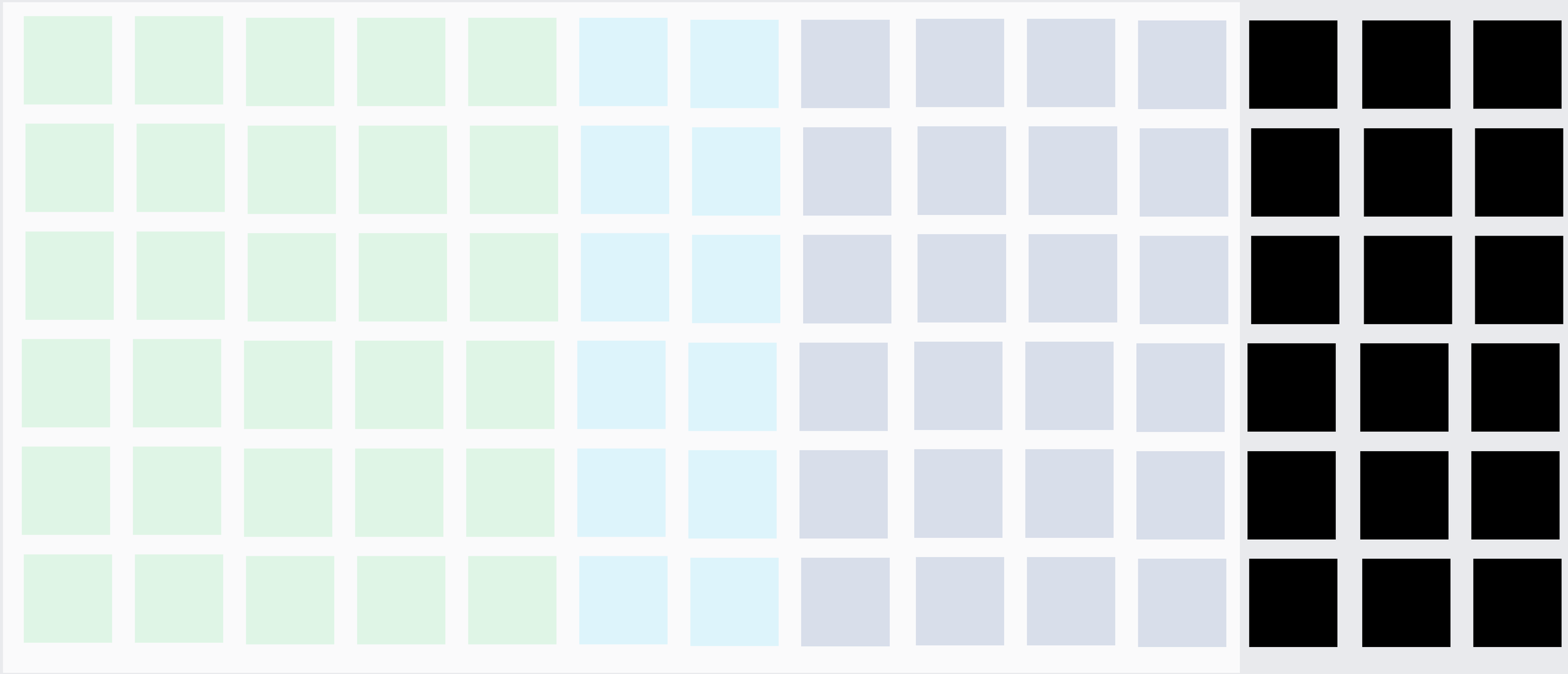
Survivor Space



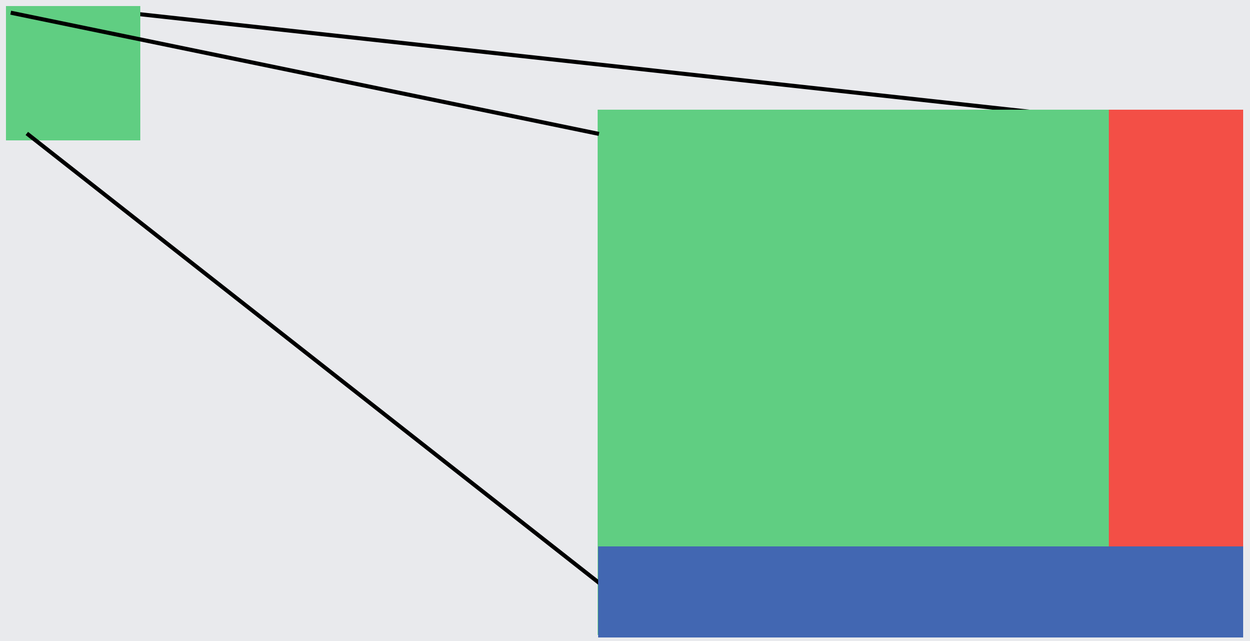
Tenured Space



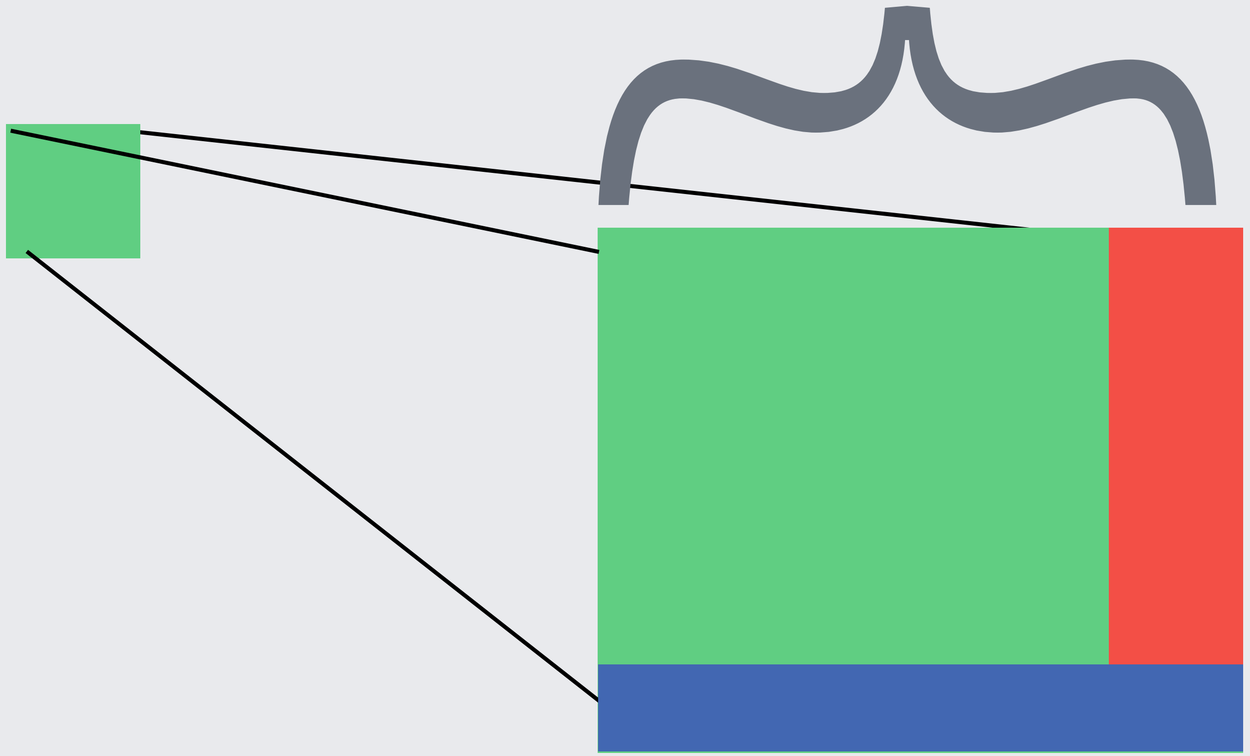
Free Space



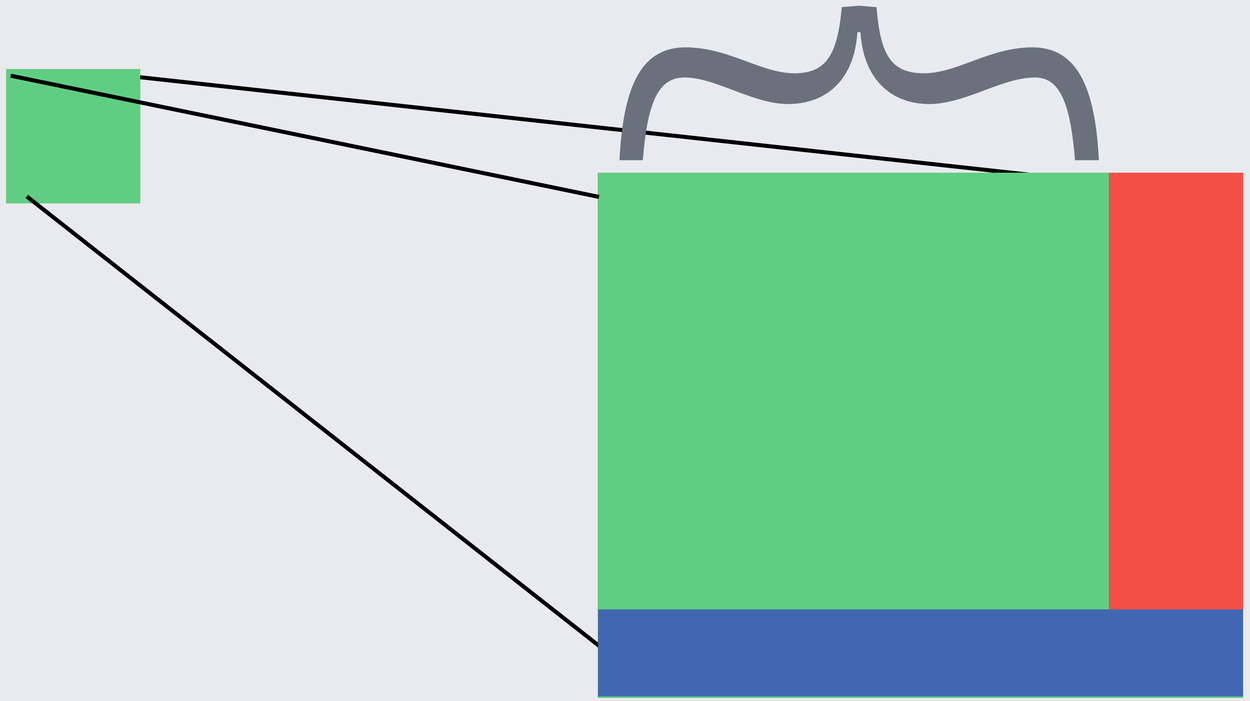
The Structure of a Region

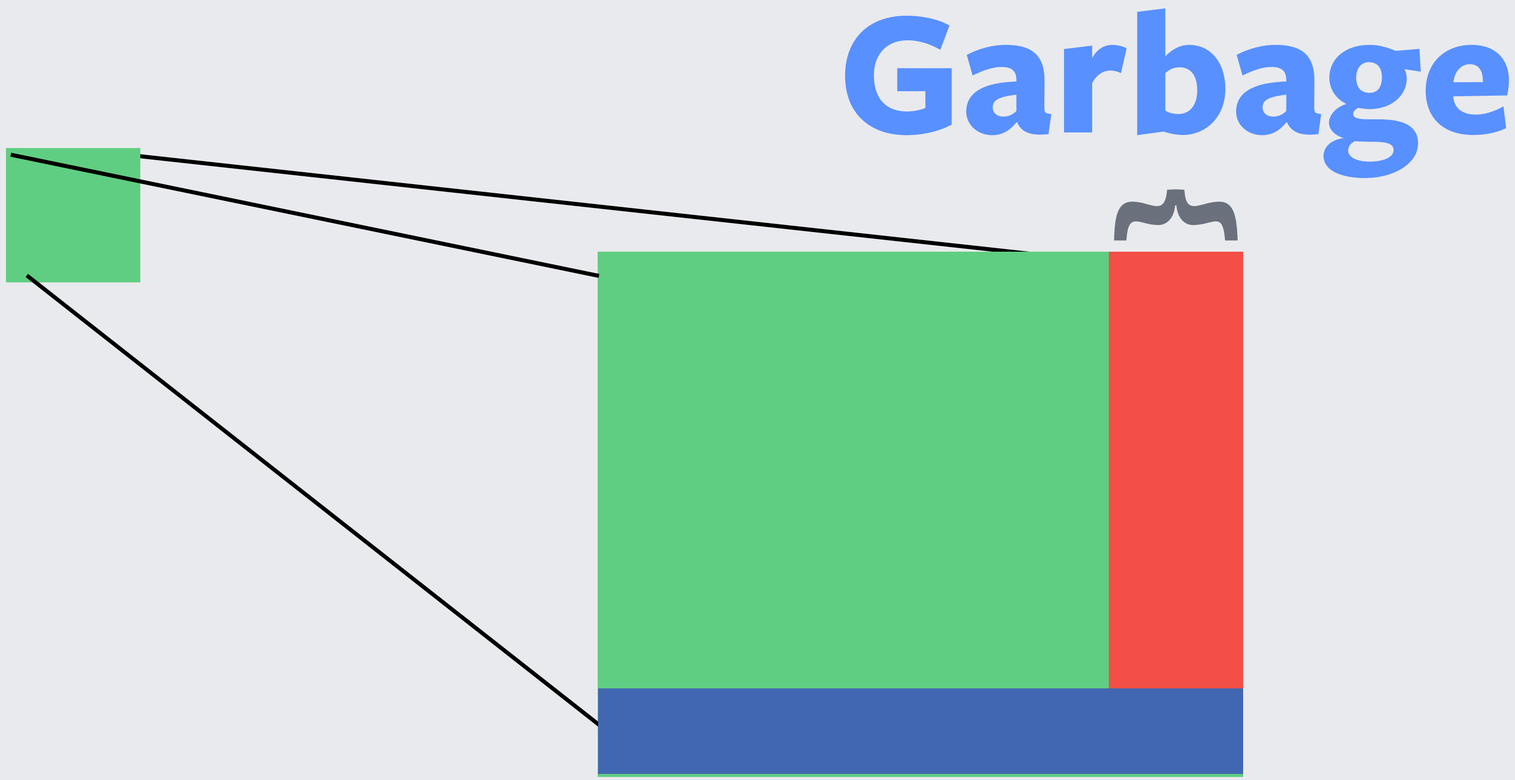


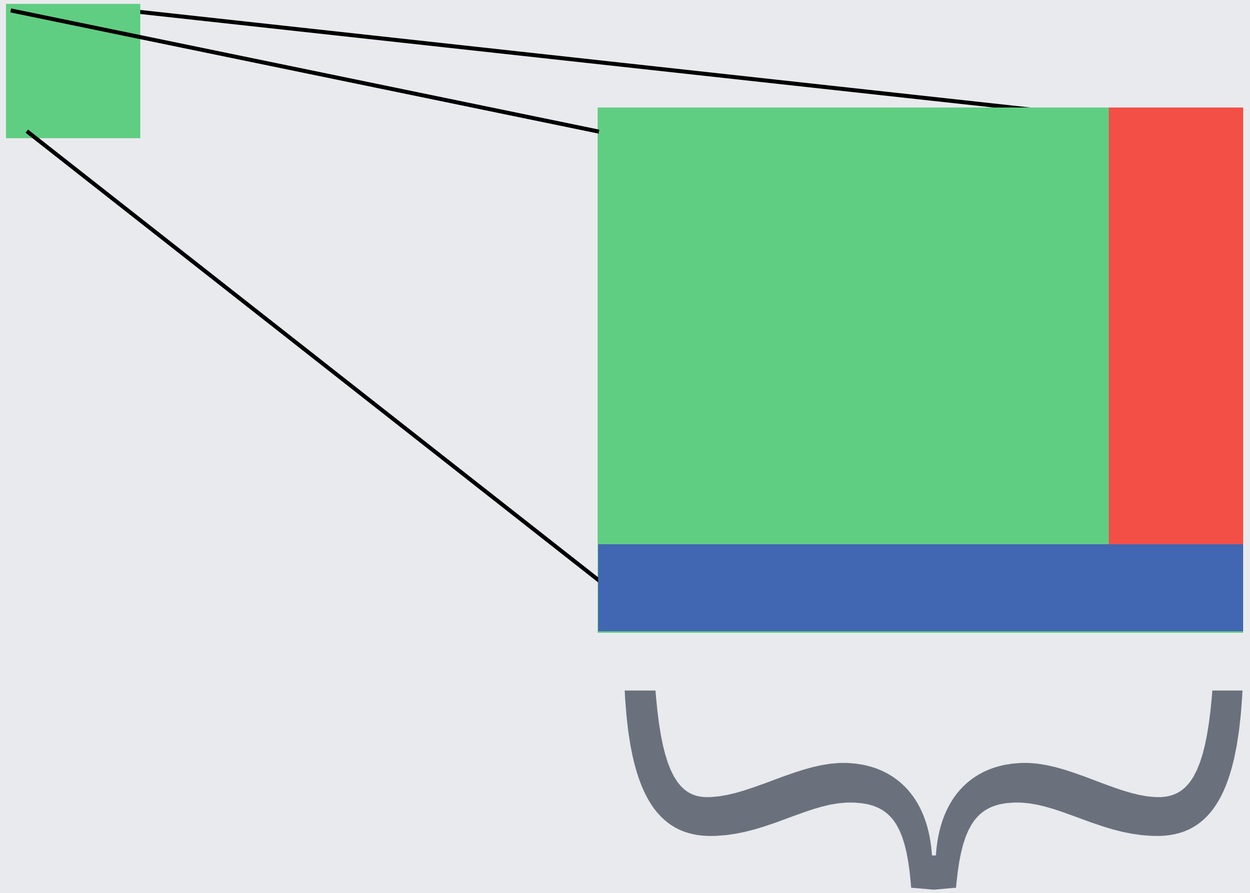
1MB <-> 32MB



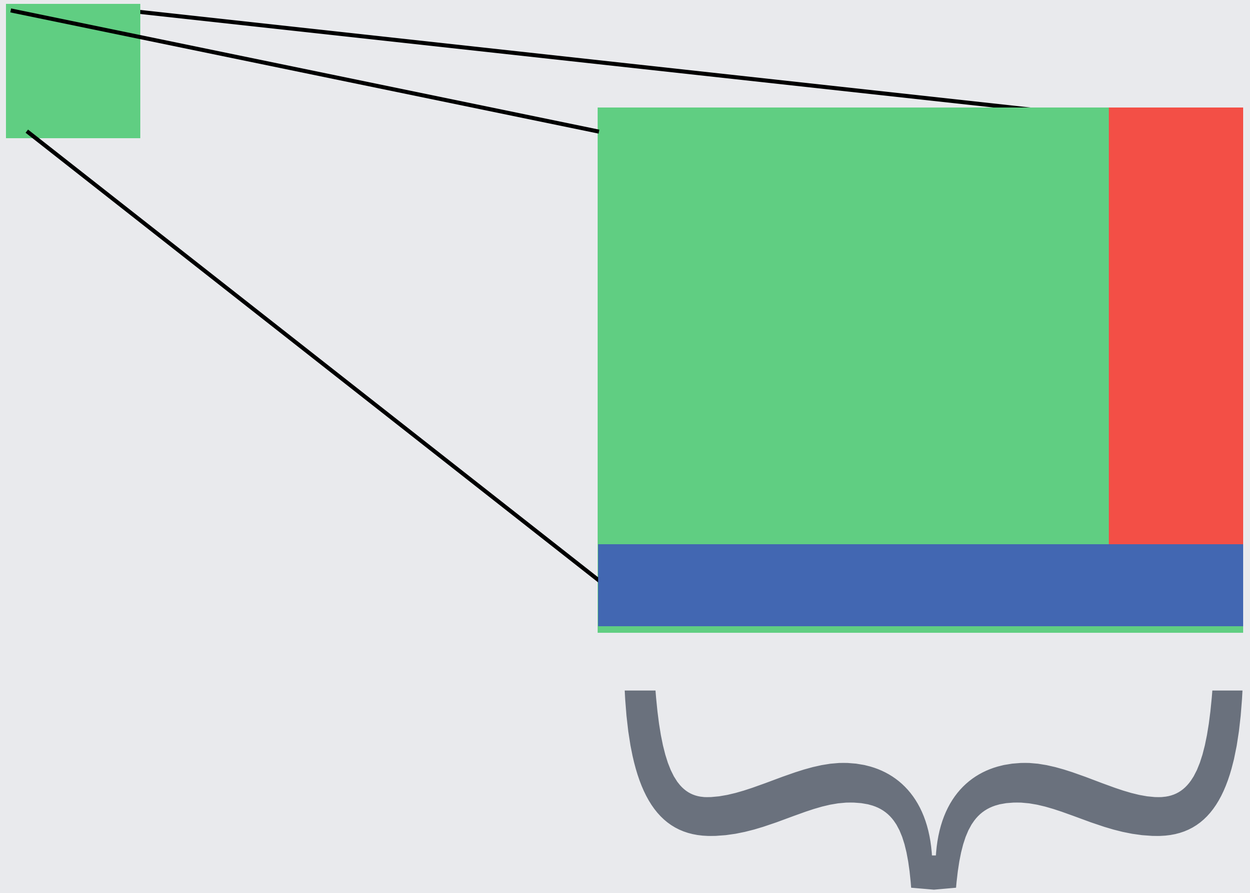
Alive




Garbage

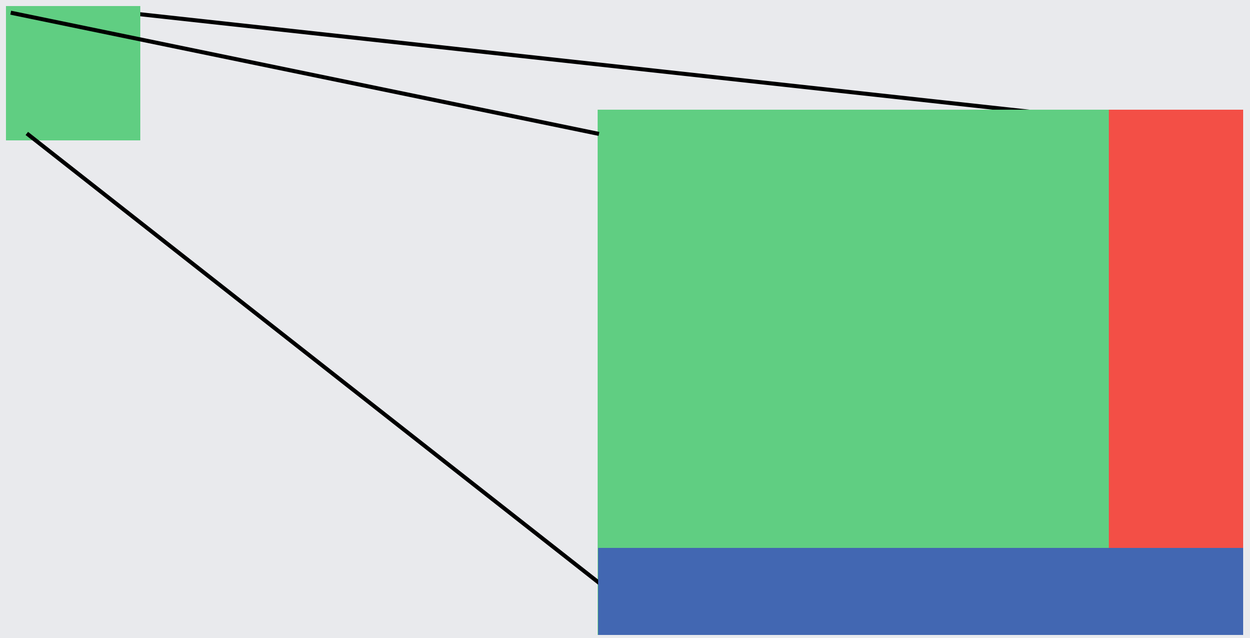


Remembered Set



Knows which region segments are alive and dead

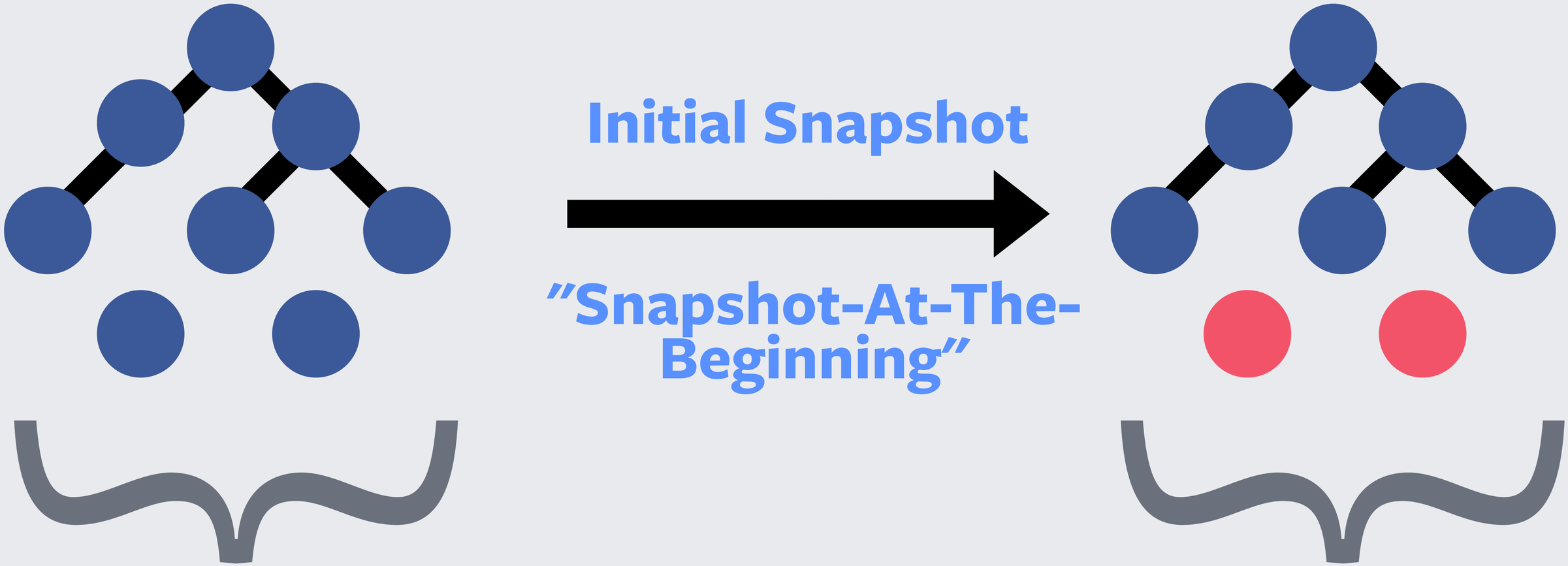
Regions have ”liveness percent”



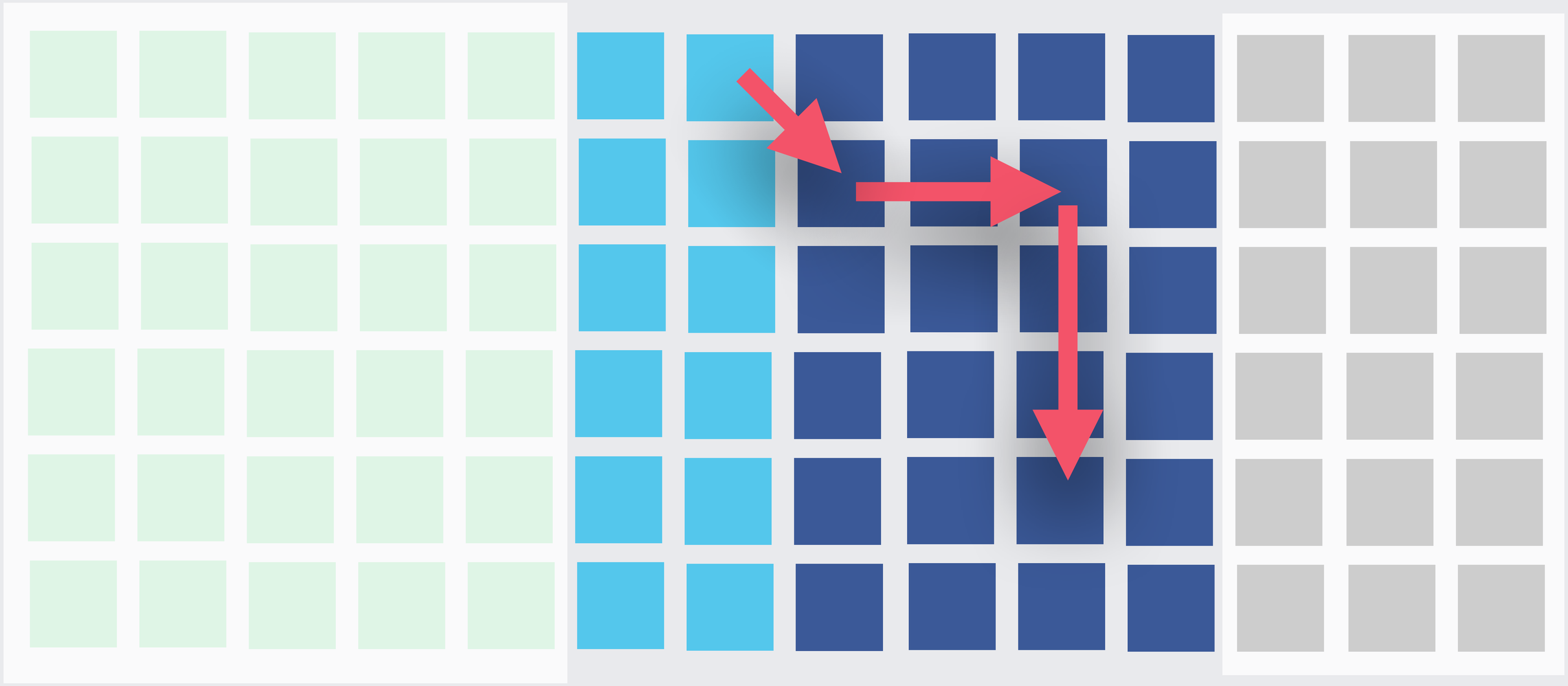
liveness_pct = live_size / region_size

G1 GC Lifecycle

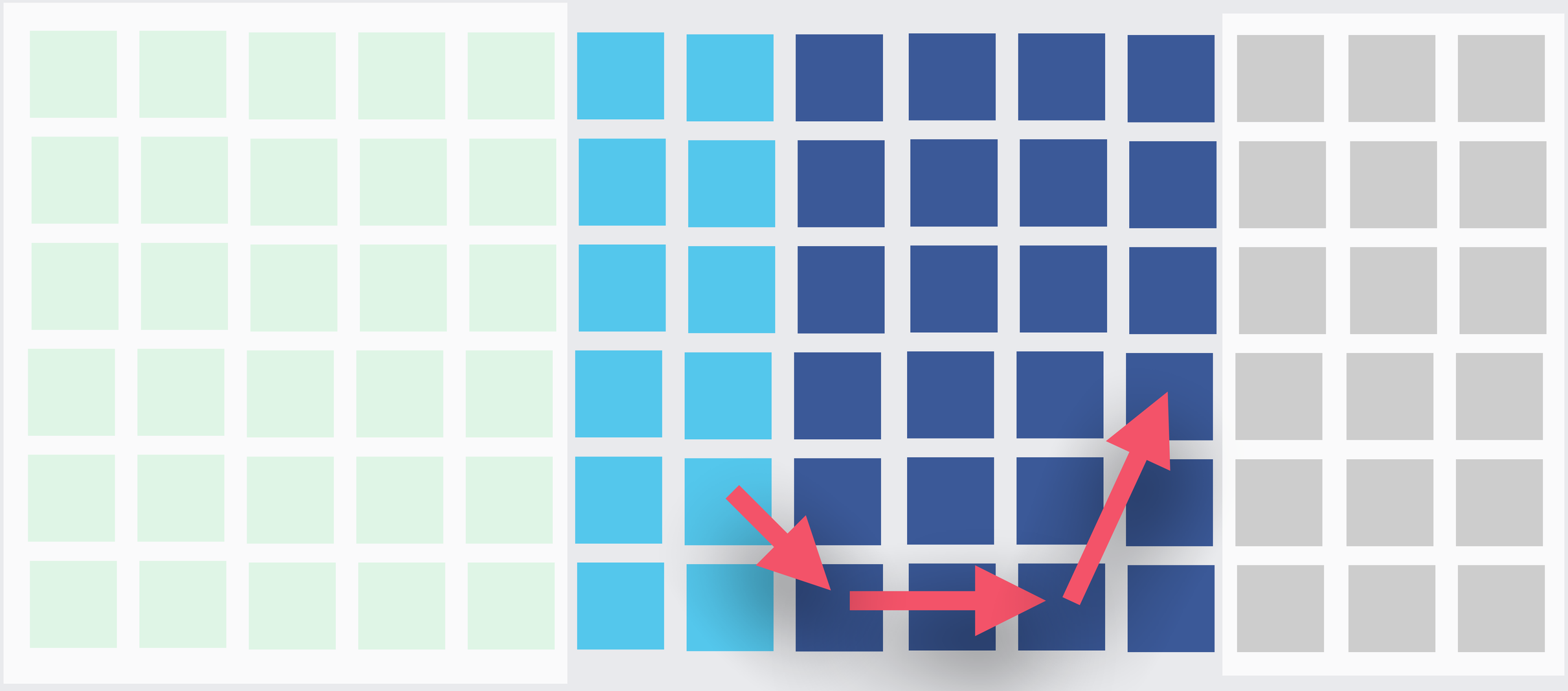
Marking Cycle: STW



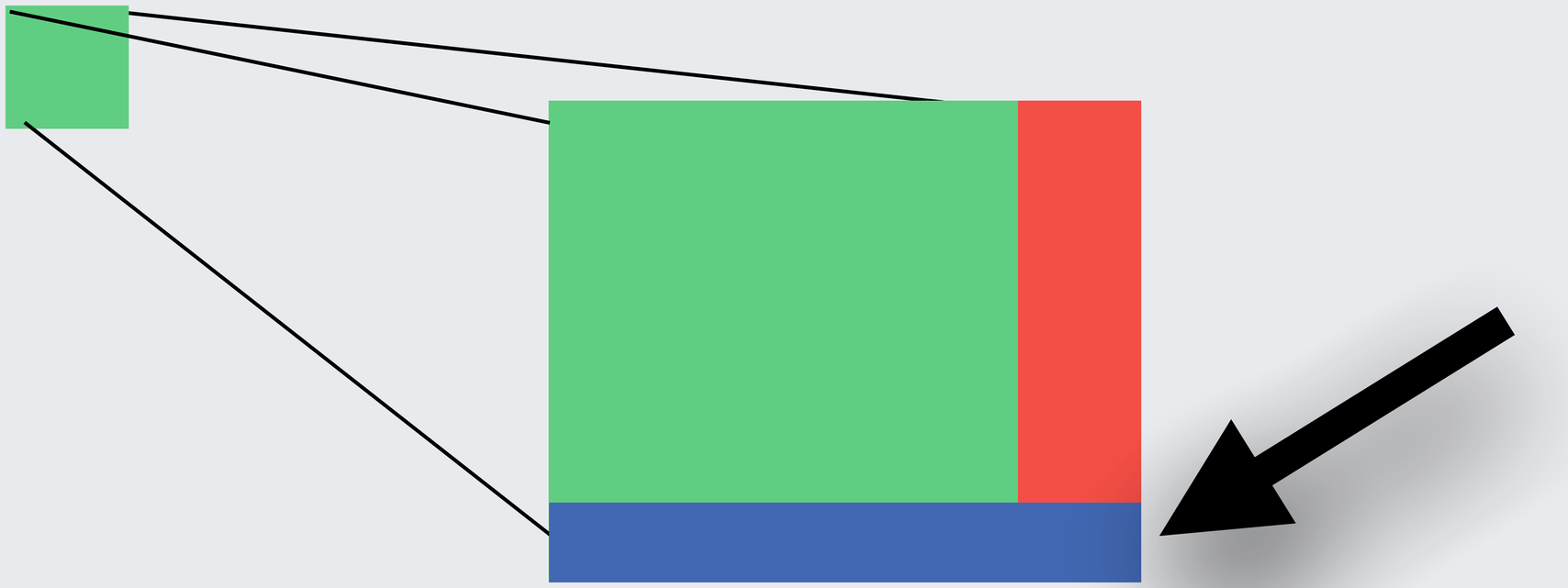
Root Region Scanning and Marking (Concurrent)



Remark (STW): Finishes Up Marking Work

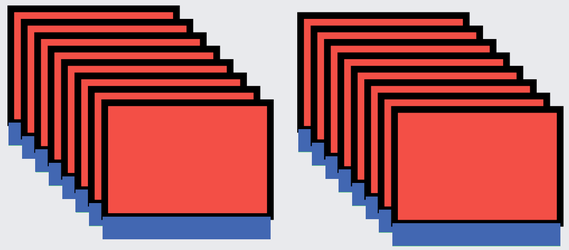


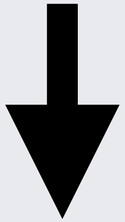
Remark (STW): Update RSet Liveness



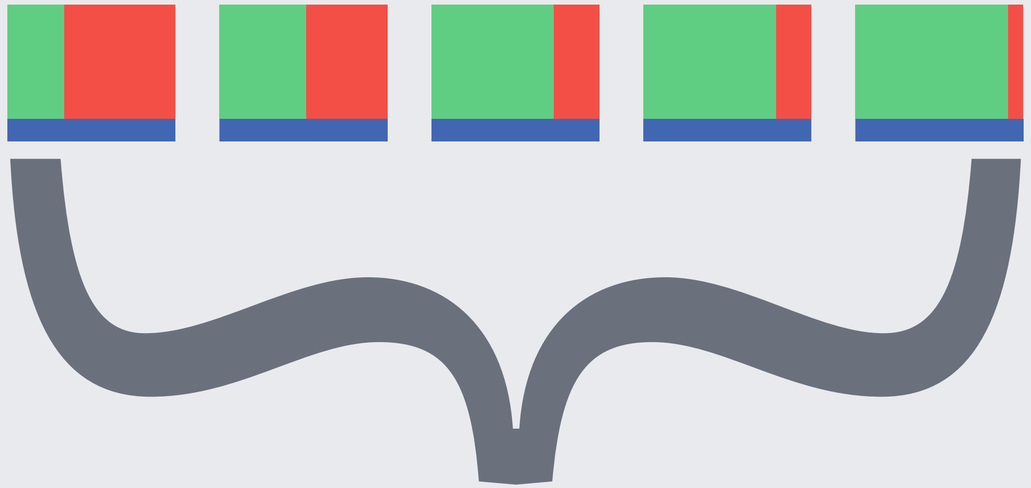
liveness_pct = live_size / region_size

Cleanup



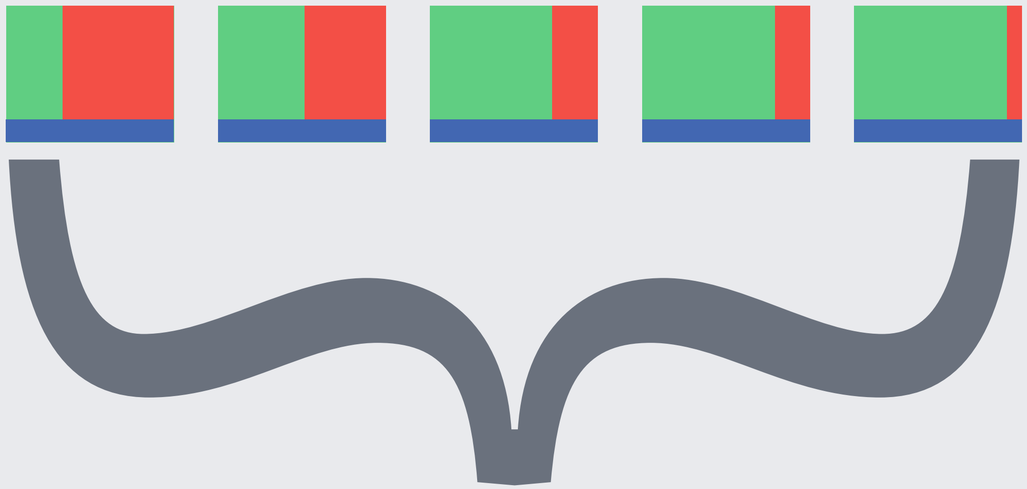


no-op to free totally empty regions

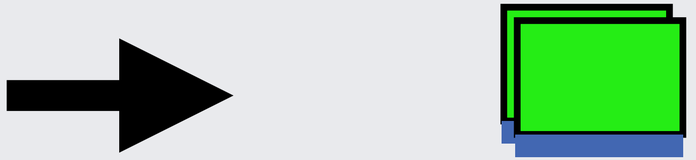


Collection Set

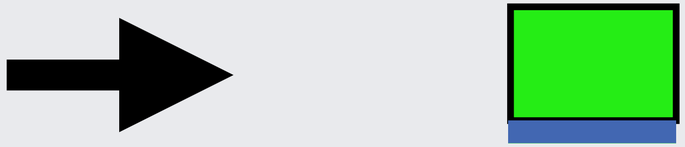
Evacuation A (STW): Young Only



Collection Set

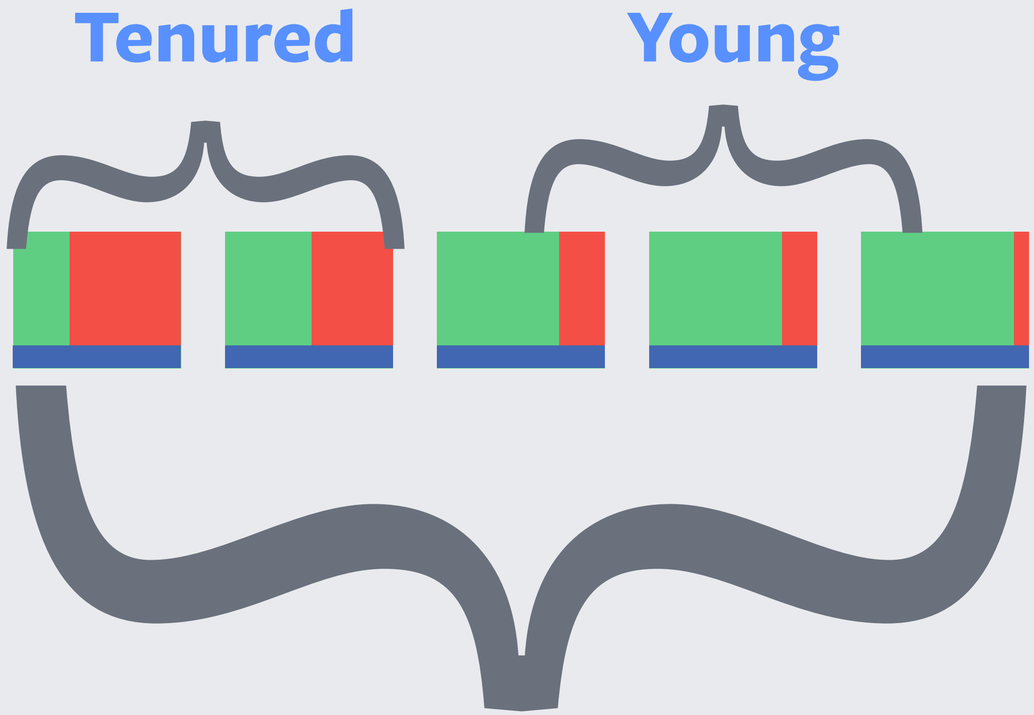


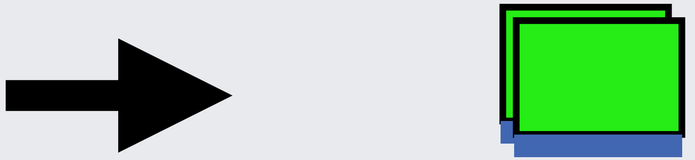
Live Objects
Copied to Survivor
Space



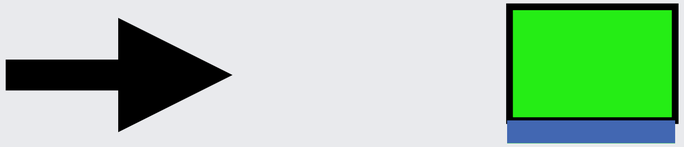
Objects that survive
beyond tenuring threshold
are moved to tenured

Evacuation B (STW): Mixed





Young Objects Copied to Survivor Space or are promoted to tenured

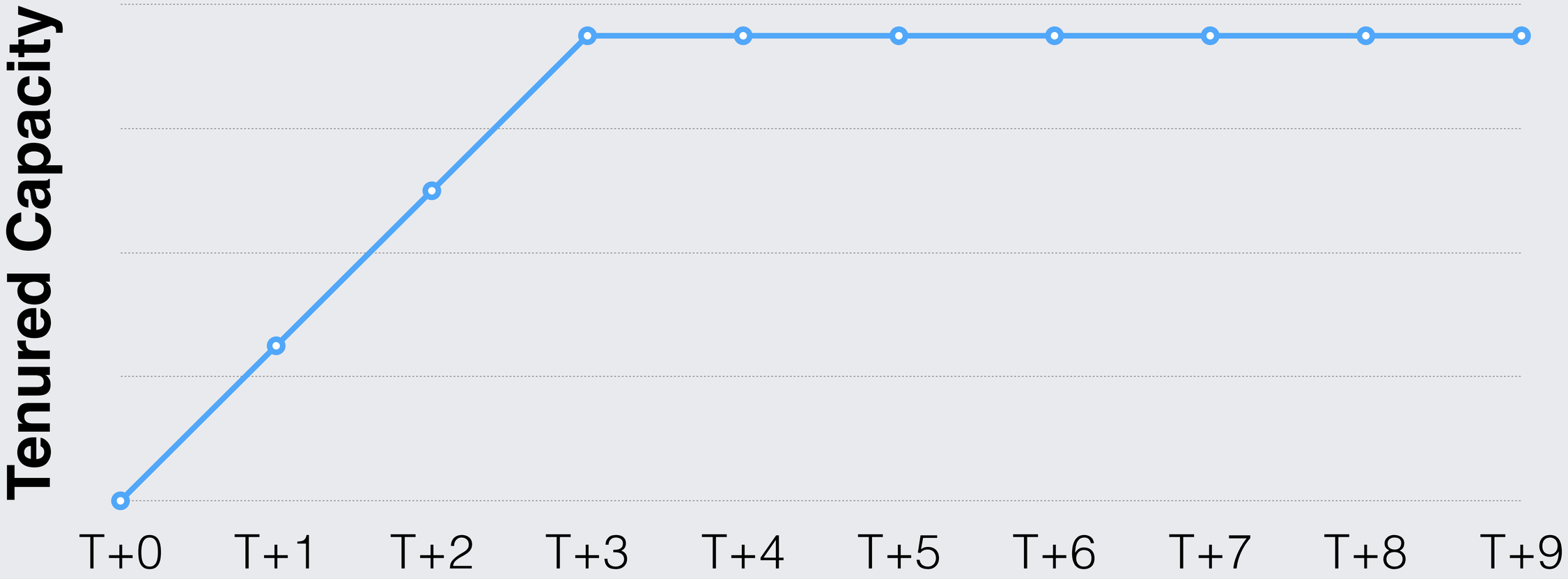


Tenured objects are copied out to new contiguous blocks in the tenured heap

Collection Set

What does ”tenured” really mean?

Tenured means ”Long Lived”



Young vs Tenured is a False Dichotomy!

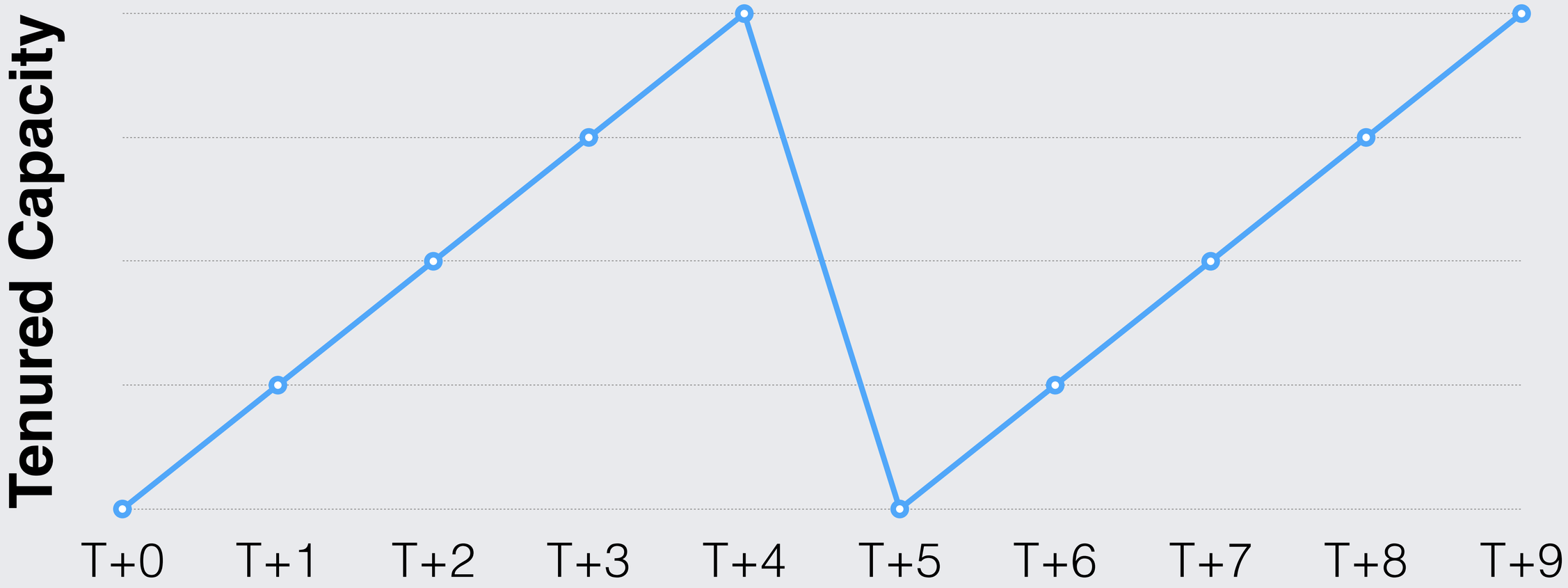
G1 GCs Implicit Assumption: Liveness

- Long-Lived
- Short-Lived

Realtime Java Service Reality

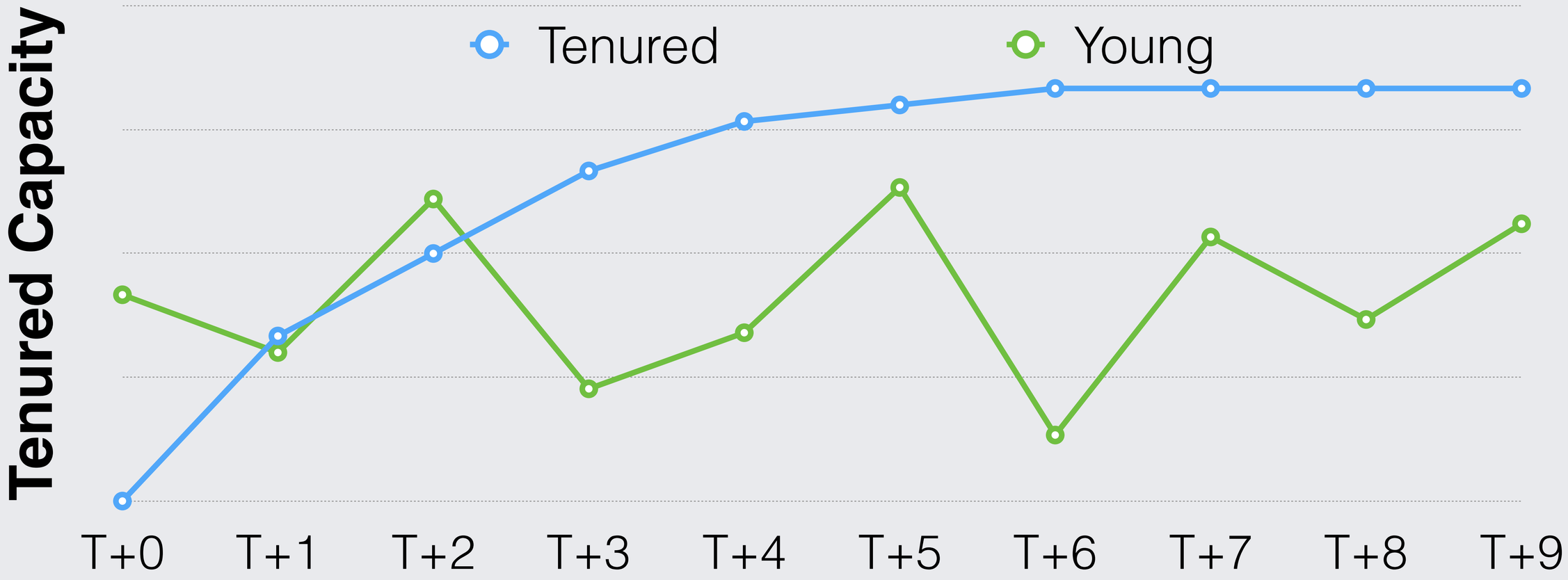
- Long-Lived
- Medium-Lived
- Short-Lived

G1 GC Defaults Antipattern: Medium Lived Objects

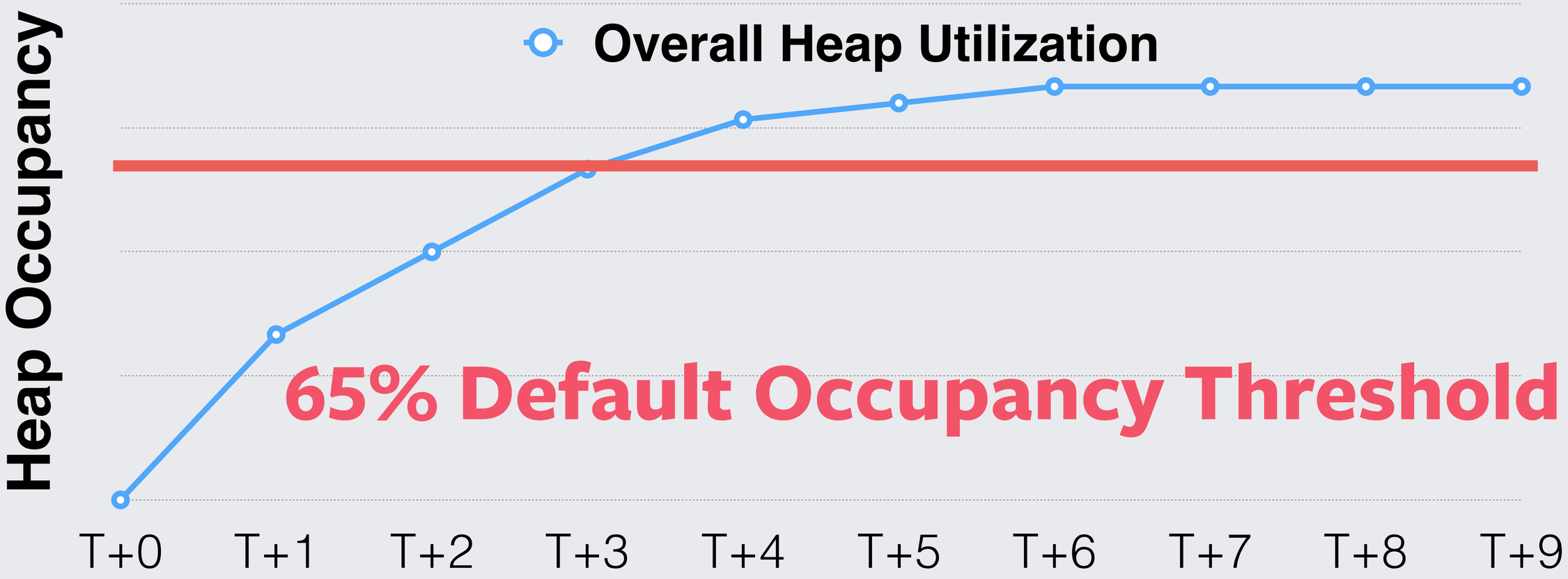


Making G1 Work: Young vs Tenured

What Should Your Heap Occupation Graph Look Like?



Mixed GCs: Occupancy Threshold

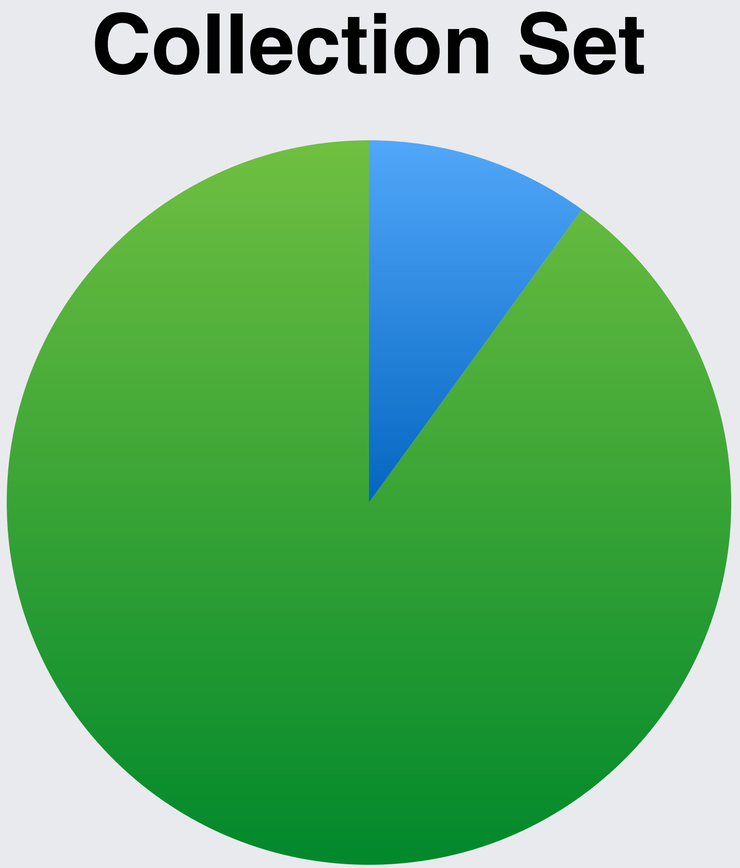


Mixed GC Default Assumptions

Collection Set Candidates Reclaim Max 10% Old Regions

Tenured

Young

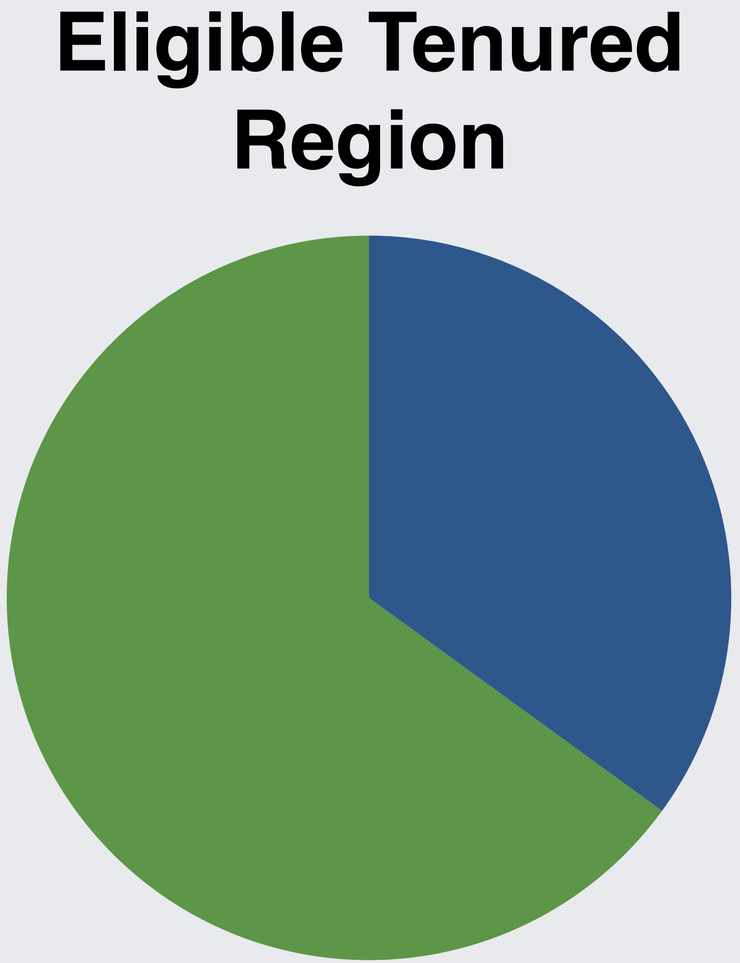


Mixed GC Default Assumptions

Eligible Collection Regions Must Be 65% Live or Below for Inclusion

Garbage

Alive



Tuning for Constant Tenured Cleanup

JVM Options

XX:G1OldCSetRegionThresholdPercent=<%>

Increase the default of 10% to a higher value to ensure more tenured regions are included in mixed GC

JVM Options

XX:InitiatingHeapOccupancyPercent=0

Force G1 to begin mixed collections immediately.

JVM Options

XX:-G1UseAdaptiveIHOP

Prevent G1 to adapt the initiating occupancy percent so that we are constantly performing mixed GCs.

JVM Options

XX:G1MixedGCLiveThresholdPercent=90

Expand the eligible tenured regions for collection during a mixed GC.

JVM Options

XX:G1MixedGCCountTarget=4

Reduce the number of mixed GC events between each young-only GC event. This ensures we purge young regularly, as well as updating the liveness percentile for tenured regions regularly to increase GC efficacy.

JVM Options

XX:MaxGCPauseMillis=400ms

Select a reasonable, achievable pause time goal that meets your garbage eviction throughput requirements. Real-time doesn't mean instant, but it does mean relatively quick and narrow variance boundaries.

Realtime Considerations

G1: Realtime Hard Requirements

- **Keep tenured generation pruned as fast as objects die**
- **Ensure there's enough heap space to absorb bursts of new short, medium, and long lived objects**
- **Leverage G1s adaptive tenuring threshold to absorb traffic spikes**
- **Avoid full GC invocations at all costs**

facebook