Incremental Copying Collection with Pinning (Progress Report)

Daniel Spoonhower
Carnegie Mellon University
spoons+@cs.cmu.edu

(Joint work with Guy Blelloch and Robert Harper)

January 12, 2004

What? Why?

- Real-time memory management
  - explicit
  - static analysis (*e.g.* region inference)
  - modern garbage collection ⟵ *this talk*
- Real-world constraints
  - *e.g.* object pinning
- Tradeoffs in GC design
  - copying vs. non-copying collection

Use *mostly-copying* collection to balance competing design goals

Background

- Extension of Cheng *et al.* work [PLDI ’01]
  - bounds time and space consumed by GC
  - minimum mutator utilization
  - based on copying collection
- Real-world environment – Rotor (a.k.a. SSCLI)
  - JIT + run-time + GC
  - pinned objects, finalizers, &c.
- Goal:
  - single framework supporting both performance and semantics

Mostly-Copying Collection

- Bartlett [TR ’88]
  - ambiguous roots (*i.e.* untyped stack values)
- Pinned objects are “uncooperative”
  - only roots are pinned
- *Mostly*-copying collection
  - heap divided into *pages*
  - *from-* and *to-space* defined logically
  - ambiguous/pinned roots promoted “in-place”

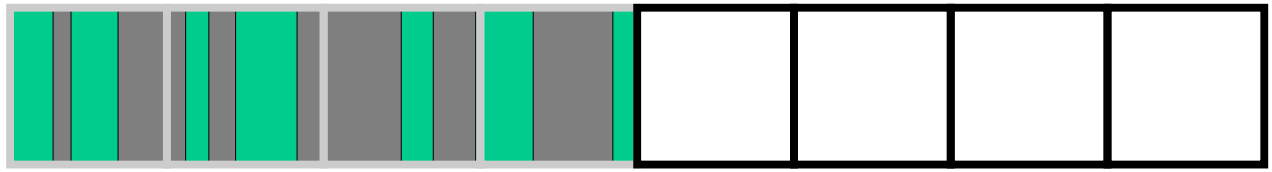
Mostly-Copying Collection

divide heap into pages. . .

Mostly-Copying Collection

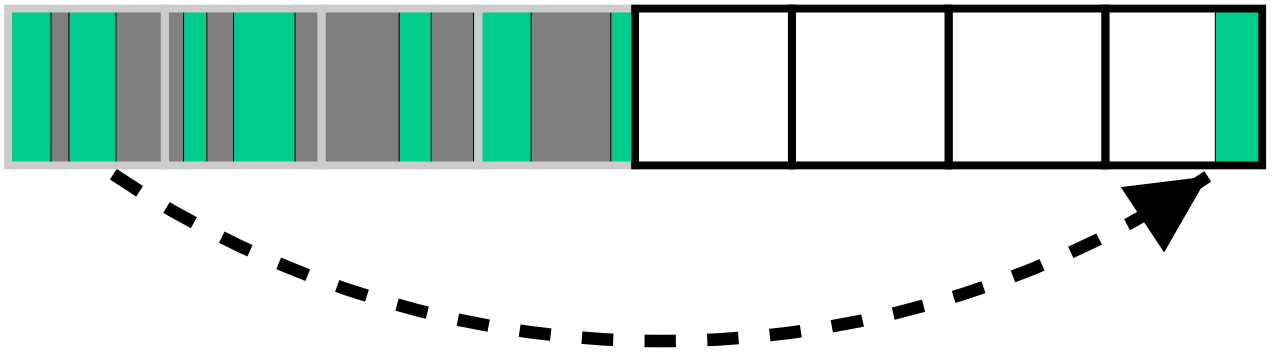
. . . allocate. . .

Mostly-Copying Collection



. . . begin collection. . .

Mostly-Copying Collection



. . . promote (by copy). . .

Mostly-Copying Collection

. . . pinned. . .

Mostly-Copying Collection

. . . page promotion!

Tradeoffs

- Copying collectors
  - simple / fast allocation
  - better asymptotic time
  - may improve locality
- Non-copying collectors
  - conservative collection
  - pinning
  - less space
  - large and older objects

Other Applications

- Large objects
  - occupy one or more pages
  - expensive to copy, often long lived
  - promote in-place
- Dense pages
  - many reachable objects
  - little fragmentation
    $\Rightarrow$ little to be gained from compaction

Page Residency

(or *Density* or *Occupancy*)

$=$ % of page that is reachable

- Estimation
  - heuristic: measure during previous cycle
    - compacted pages $\rightarrow$ 100%
    - promoted pages as measured
    - young pages $\rightarrow$ 0%
- Residency threshold
  - determines when to promote by copy / in-place
  - causes behavior to range from semi-space to mark-sweep

Preliminary Results

- Effectiveness of promotion strategy
  - fraction of promotion in-place
  - error in estimate (as % of in-place)

Benchmark	Page Promoted	Estimate Error
huffman	90.03%	0.04%
xml	51.89%	10.36%
splay	70.25%	11.86%

Continuing Work

- Continued analysis
- Experiments
- Incremental, concurrent, parallel collection
- Impact of other language features on GC