A Page Fault Equation for Dynamic Heap Sizing

me

Y.C. Tay

National University of Singapore

X.R. Zong

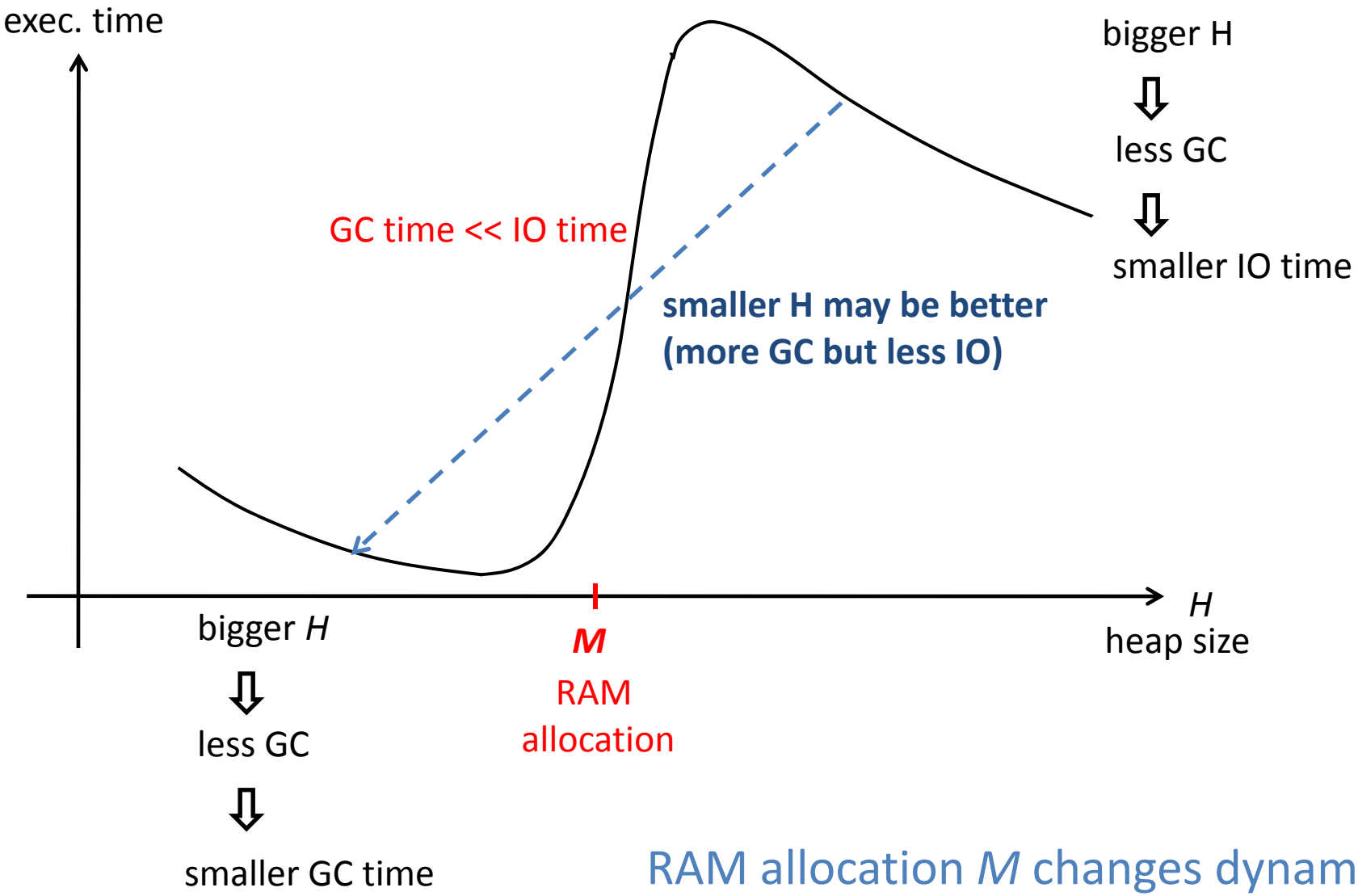
Duke University

Java, C#, Ruby … require garbage collection (GC) for the heap

heap size H determines #GC

issue: how to tune H ?

For a garbage-collected (GC) application,
how does execution time vary with heap size *H* ?

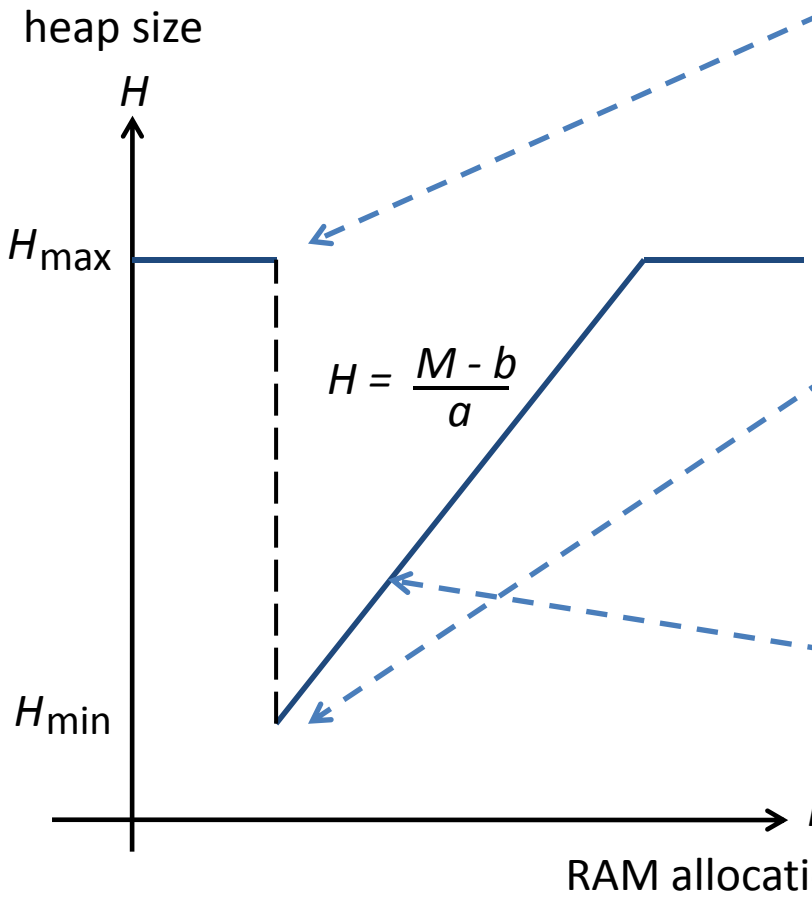


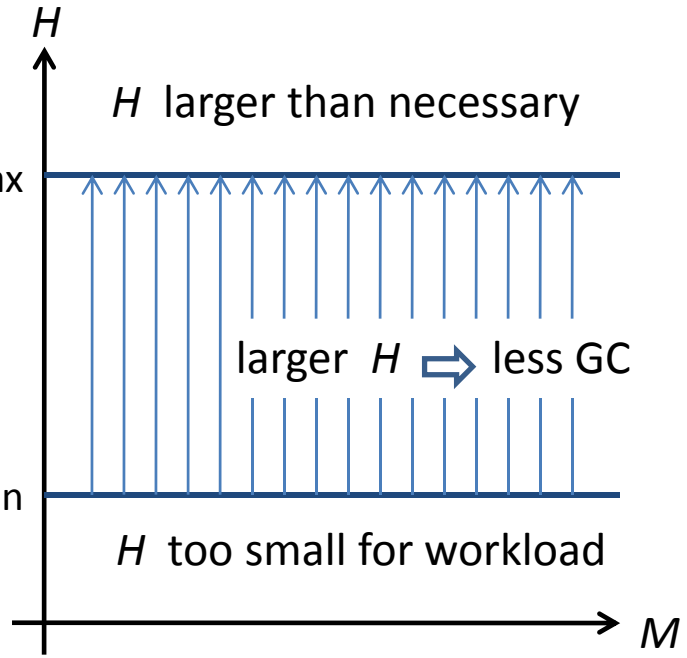
RAM allocation *M* changes dynamically

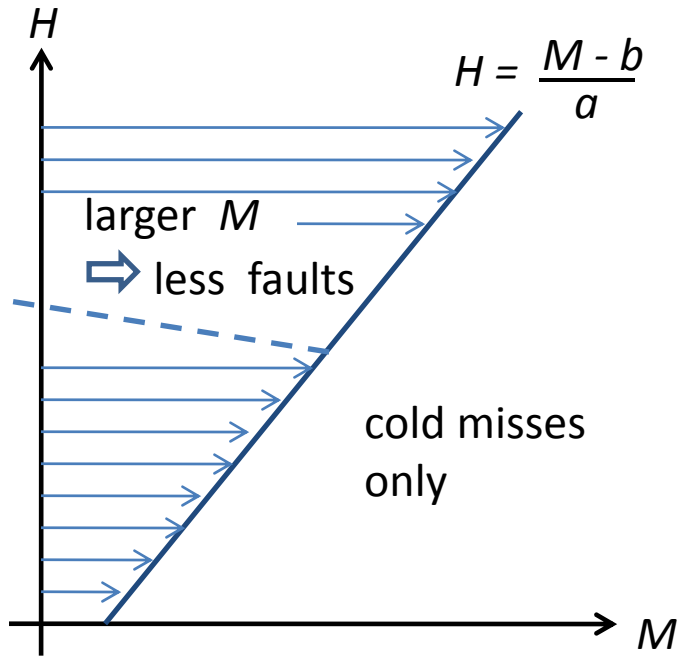
How should *H* vary with *M* ?

How should *H* vary with *M* ?

our answer: Heap Sizing Rule



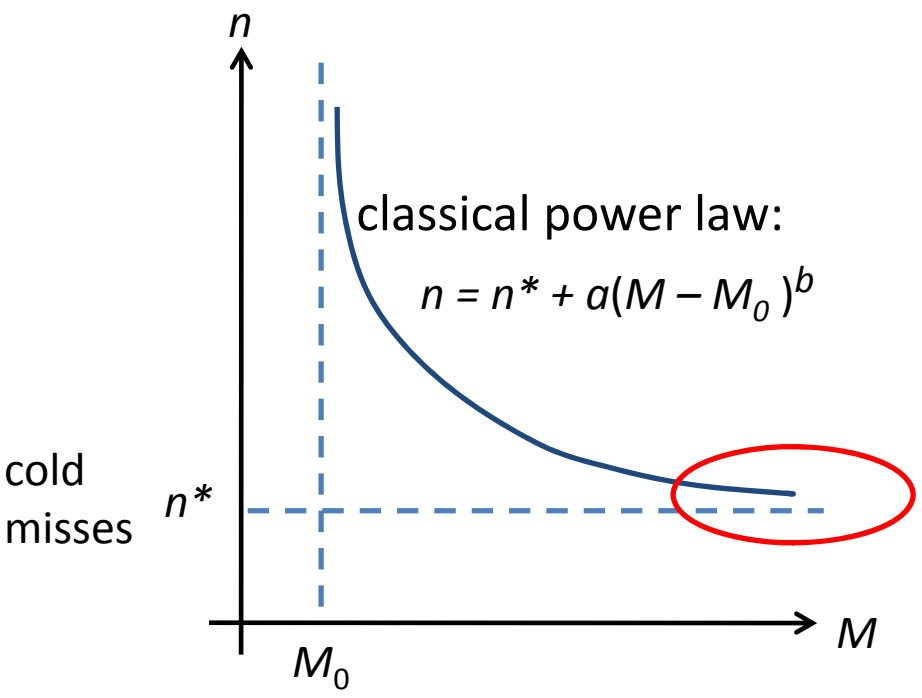




Where does $H = \frac{M - b}{a}$ come from?

Where does $H = \frac{M - b}{a}$ come from?

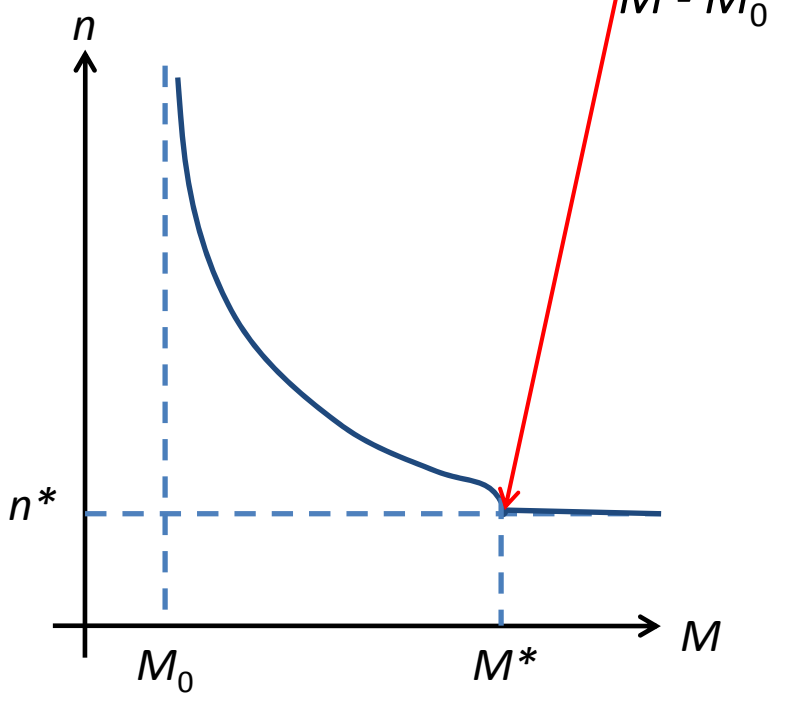
How does #pagefaults n vary with M ?



Page Fault Equation [TZ]:

$$n = \frac{1}{2}(K + \sqrt{K^2 - 4})(n^* + n_0) - n_0$$

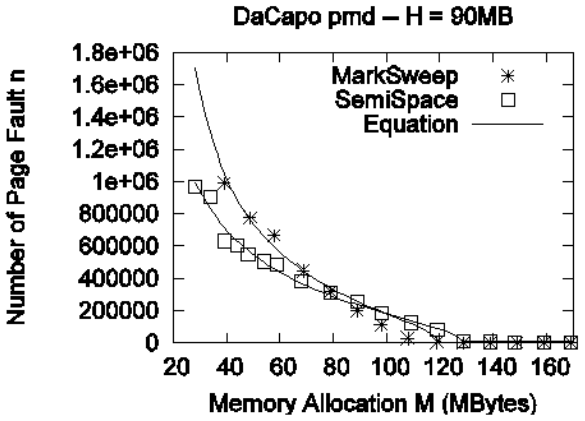
where $K = 1 + \frac{M^* - M_0}{M - M_0}$



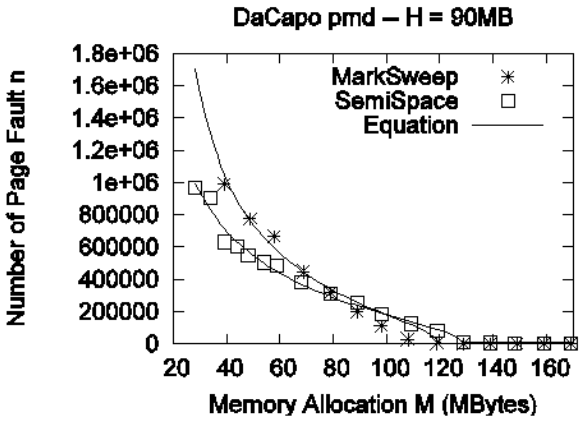
universal: works for
Linux, Windows,
compute/IO/memory-intensive workloads,
garbage-collected applications

[TZ] Tay and Zou, *A page fault equation for modeling the effect of memory size*, Perf. Eval. (2006)

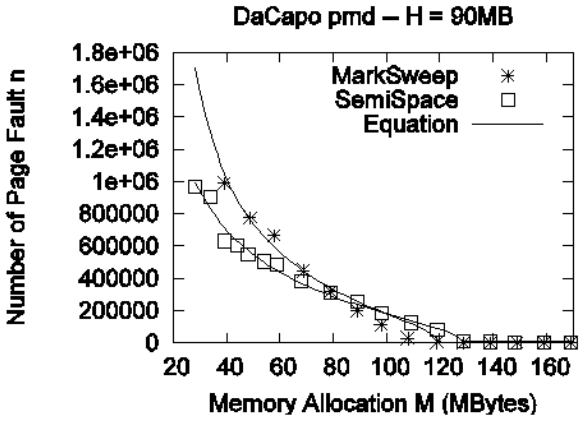
different heap sizes:



different mutators:



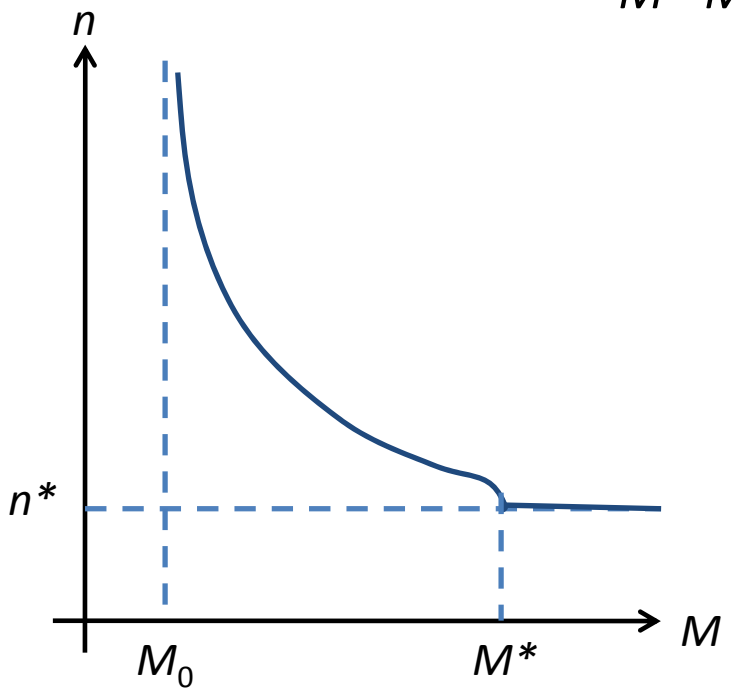
different garbage collectors:



Page Fault Equation [TZ]:

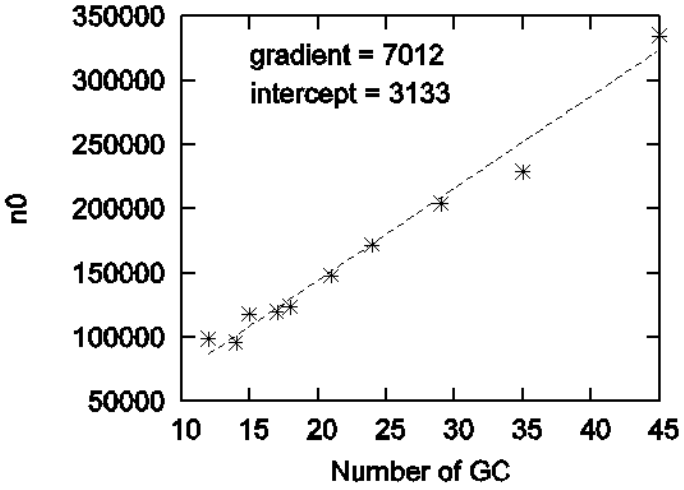
$$n = \frac{1}{2}(K + \sqrt{K^2 - 4})(n^* + n_0) - n_0$$

where $K = 1 + \dfrac{M^* - M_0}{M - M_0}$



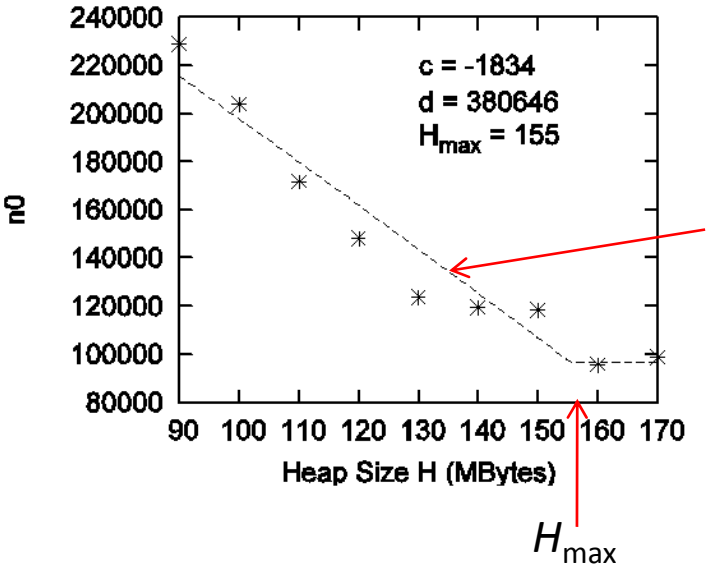
universal: works for Linux, Windows, compute/IO/memory-intensive workloads, garbage-collected applications

Interpretation for n_0 ?



n_0 measures memory taken off freelist during GC

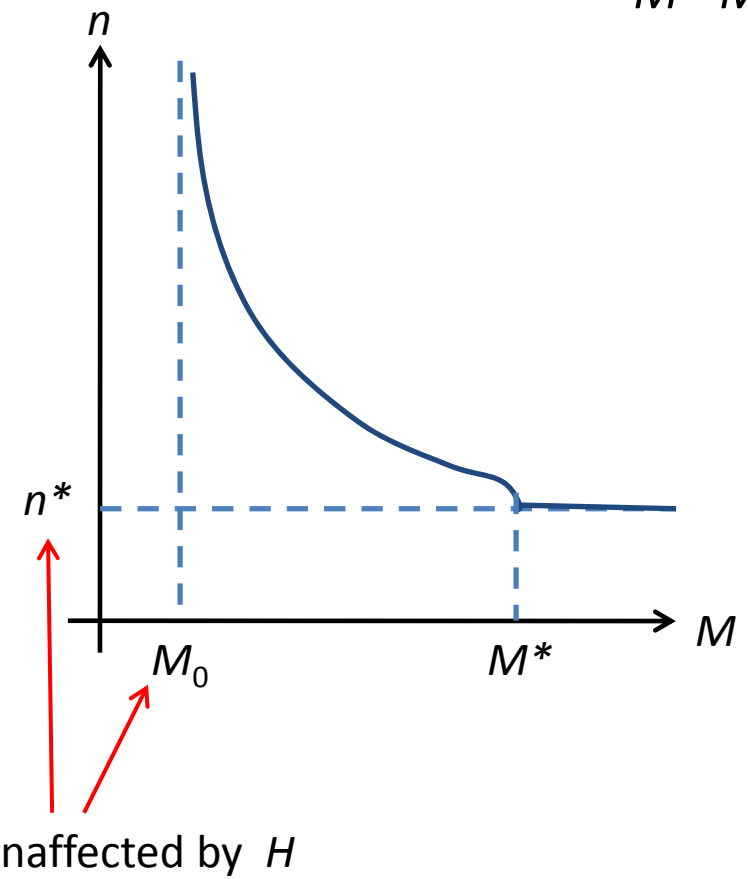
How does H affect n_0 ?



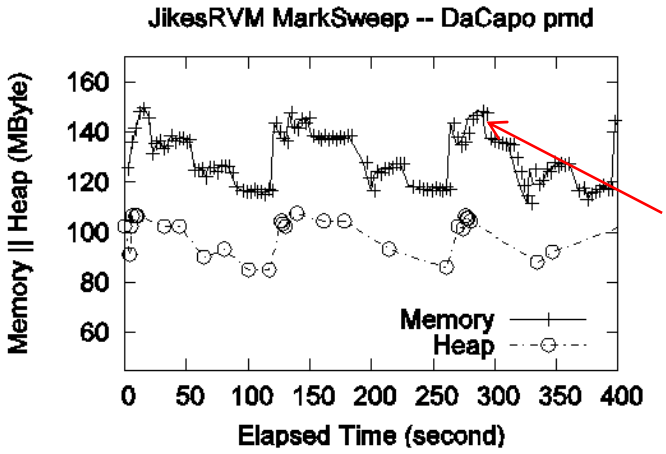
Page Fault Equation [TZ]:

$$n = \frac{1}{2}(K + \sqrt{K^2 - 4})(n^* + n_0) - n_0$$

where $K = 1 + \dfrac{M^* - M_0}{M - M_0}$



How does H affect M^*?

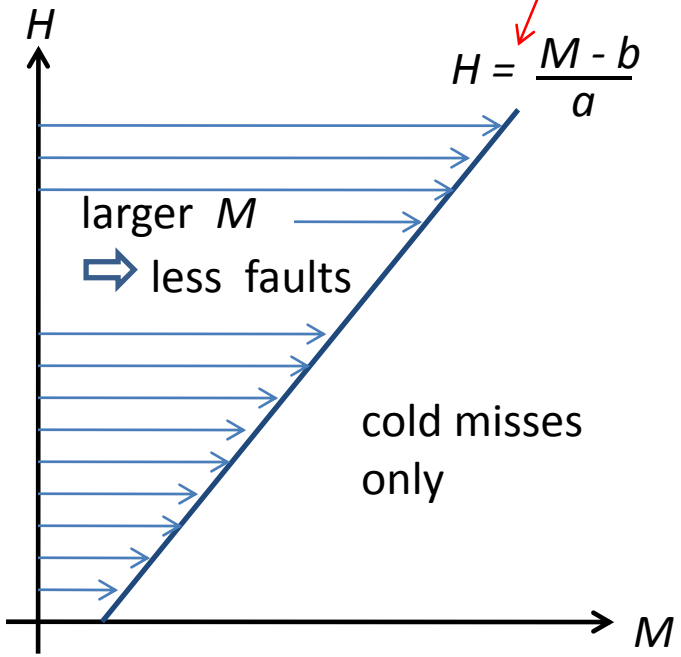


$M^* = aH + b$

depends on GC

memory overhead

recall:

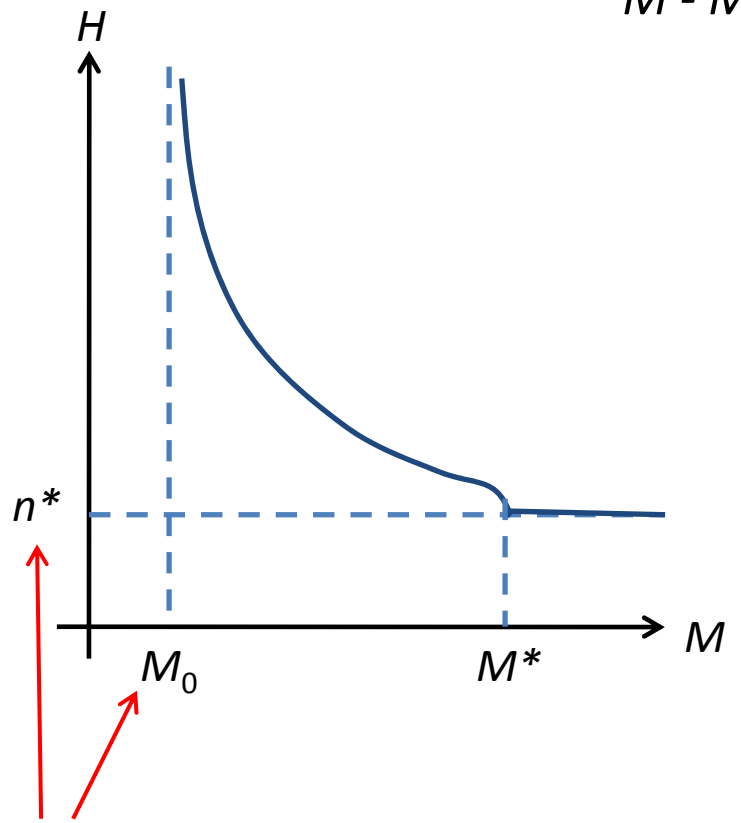


$H = \frac{M - b}{a}$

Page Fault Equation [TZ]:

$$n = \frac{1}{2}(K + \sqrt{K^2 - 4})(n^* + n_0) - n_0$$

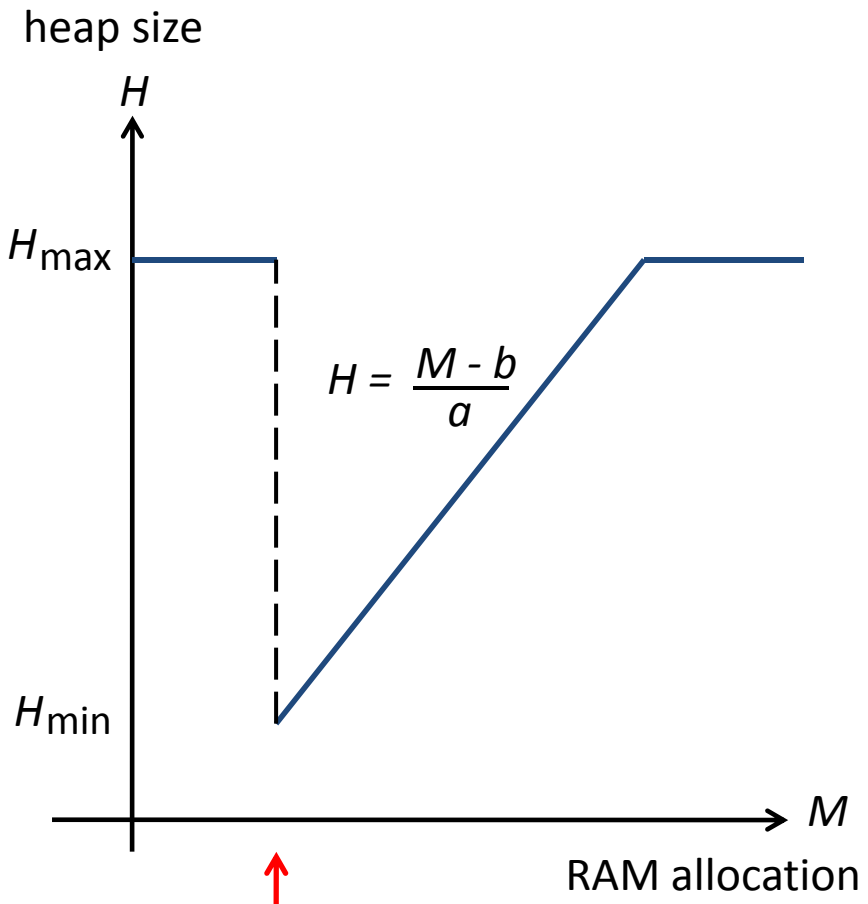
where $K = 1 + \frac{M^* - M_0}{M - M_0}$



unaffected by H

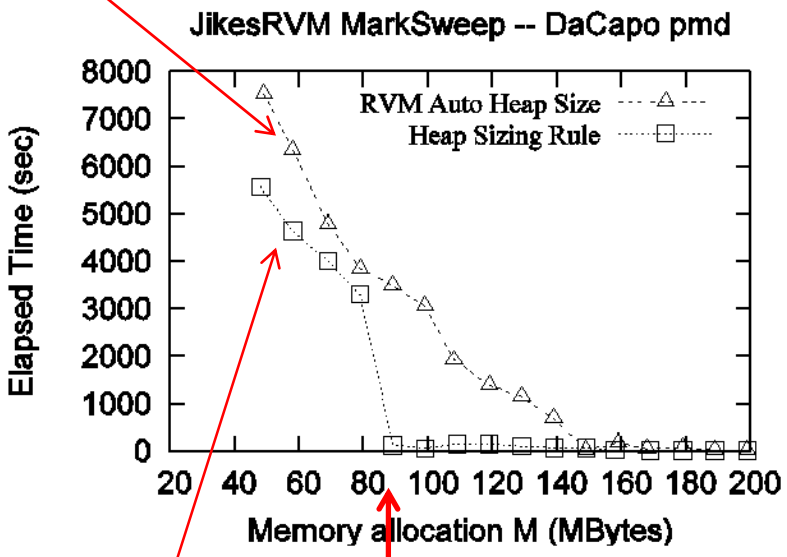
How should *H* vary with *M* ?

our answer: Heap Sizing Rule



experiment: static *M*

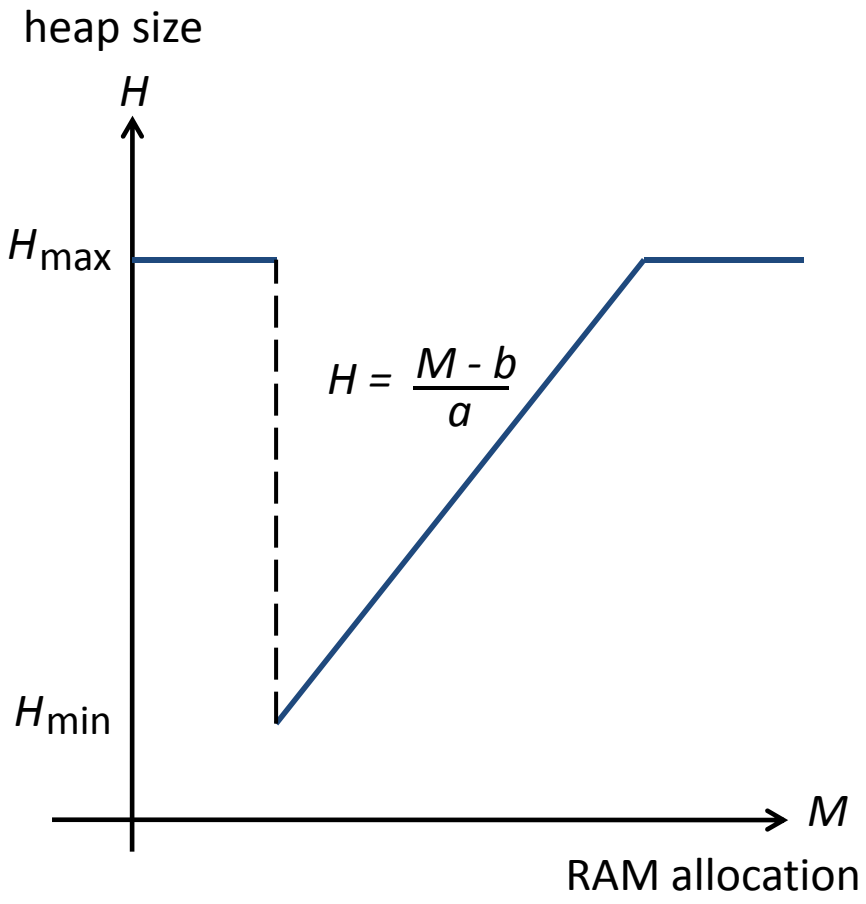
JikesRVM dynamic heap sizing
(varies *H* during execution
according to heap utilization)



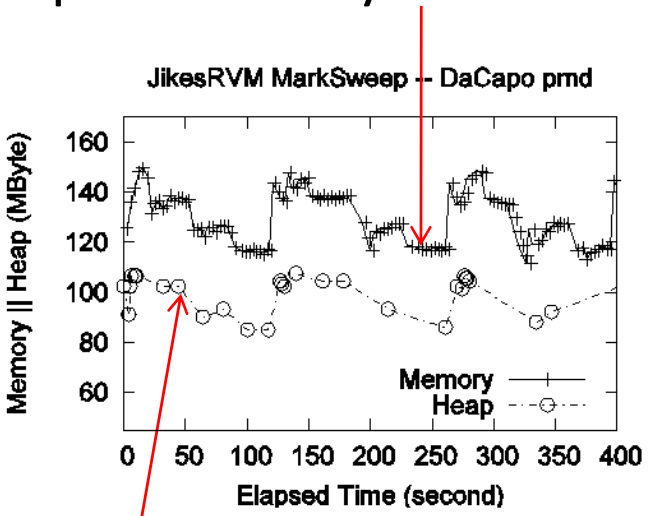
static *H* according to Heap Sizing Rule

How should *H* vary with *M* ?

our answer: Heap Sizing Rule



experiment: dynamic *M*



dynamic *H* according to Heap Sizing Rule
(*H* adjusted during GC only)

		MarkSweep pmd	SemiSpace pmd	MarkSweep xalan
page faults	RVM	425828	680575	352338
	Rule	36228	36470	64580
execution time (sec)	RVM	4762	8362	4202
	Rule	419	404	761

summary

- pagefault modeling is difficult for GC applications
  --- reference pattern changes with *H*
- our paper presents a heap-aware pagefault equation
- this equation can be used for dynamic heap sizing

future work

using the equation for heap partitioning