Non-tracing, non-conservative, non-blocking, scalable, locality-perserving, but probabilistic, garbage collector

- Kenjiro Taura

Issues addressed

- Non-blocking memory allocation
- Scalability of GC in many core machines
- Scalability of GC for large heaps (pause times)
- Cache pollution by GC's touching every reachable objects
- Leaks in conservative GCs (due to conservative pointer identification)
- Reachable but unused memory
- ABA problem

How?

- By not reusing *virtual* addresses,
- but reusing *physical* memory

- Non-blocking memory allocation
  - malloc() just bumps the allocation pointer a += sz
- Scalability of GC in many core machines
  - Our GC does not trace object graphs
- Scalability of GC for large heaps (pause times)
  - Our GC reclaims memory incrementally
- Cache pollution by GC's touching every reachable objects
  - Our GC doesn't touch objects

- Leaks in conservative GCs (due to conservative pointer identification)
  - Our GC does not follow pointers
- Reachable but unused memory
  - Our GC does not use reachability to find live objects
- ABA problem
  - Our scheme does not reuse addresses

How many bits in a virtual address?

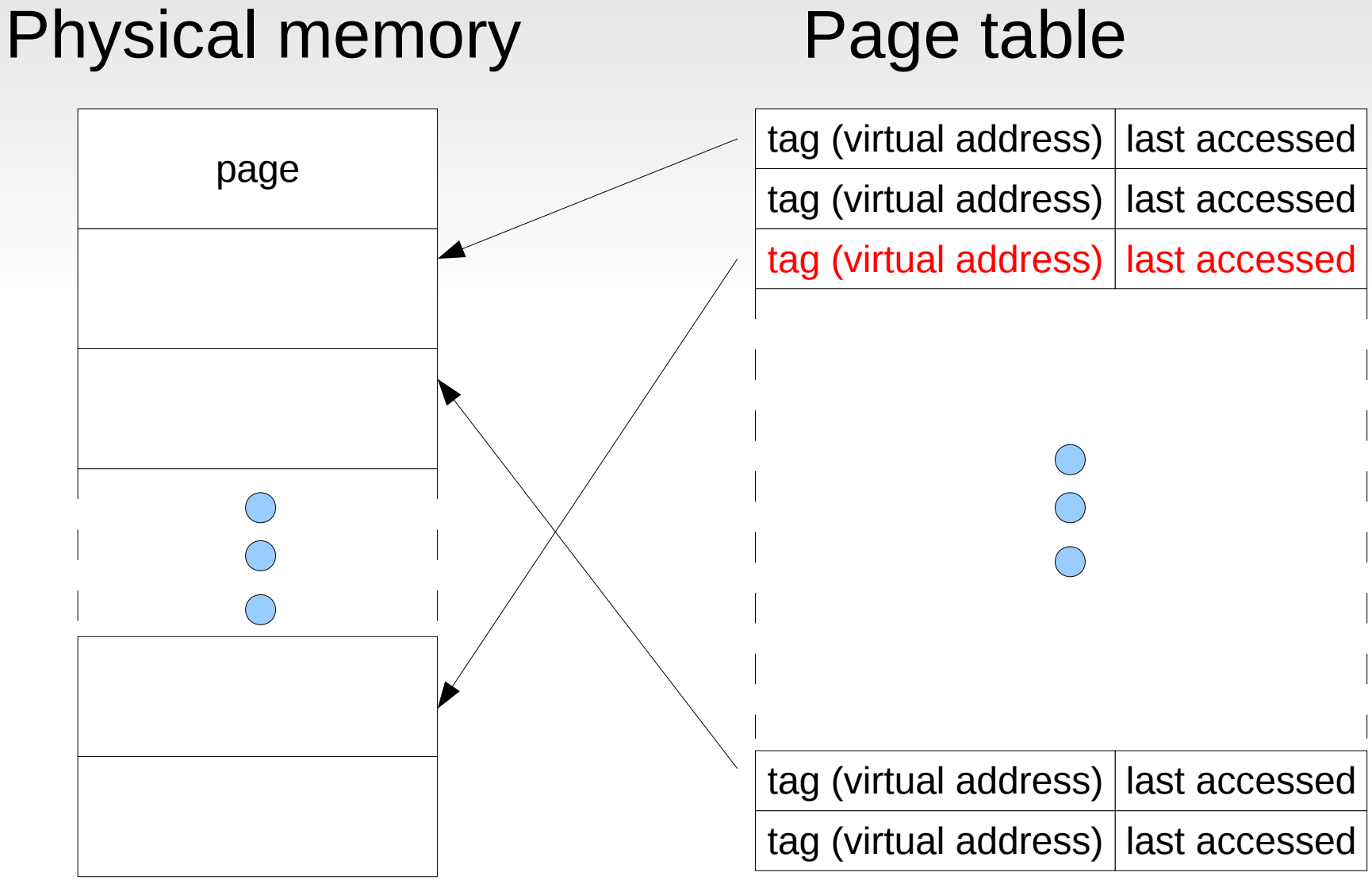
- Memory bandwidth x process lifetime
- A program running 100 years on Intel Nehalem:

32 GB/sec x 86400 x 365 x 100 = 2^68

2^35 2^17 2^9 2^7

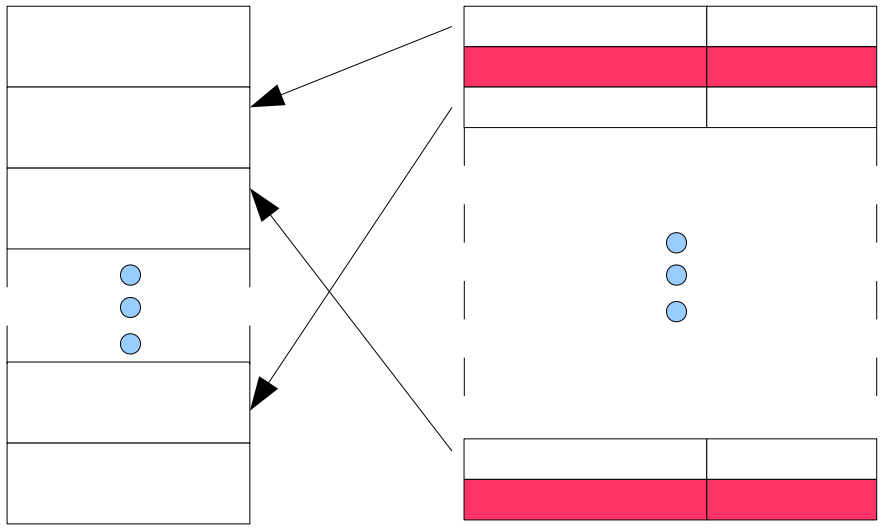
Proposal: make virtual address 72 bits

Virtual → Physical mapping structure



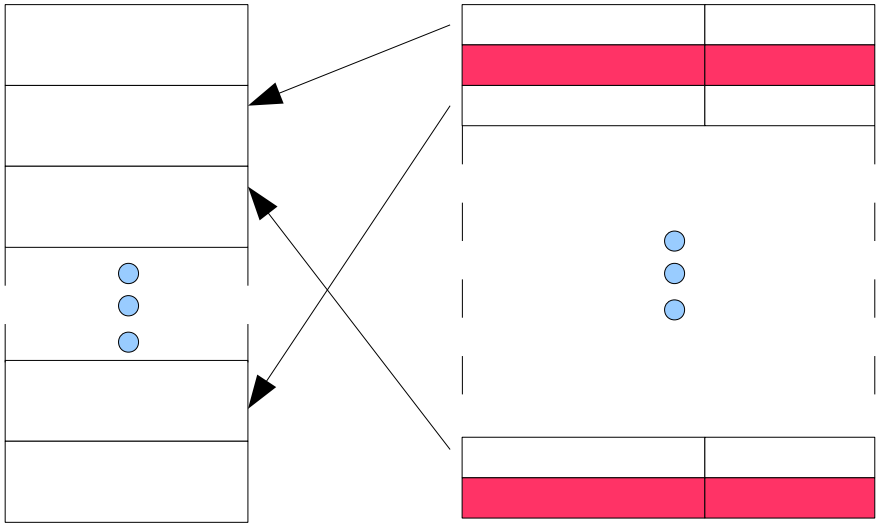
Allocating memory

- a = next address in the virtual address space;
- Apply a to K hash functions to get K hash values h_1, h_2, ..., h_K ;
- Search these K entries in the table and evict the oldest entry (reuse physical memory);



Accessing memory

- a = virtual address you are trying to access;
- Apply a to K hash functions to get K hash values h_1, h_2, ..., h_K;
- Search these K entries in the table;
- If found, access it. Otherwise you are out;



Analysis

- Assume a machine having *P* pages physical mem.
- Assume live data is <= *P* pages (for the sake of analysis)
- Assume the table has *Q* entries
- *What is the probability that at any given allocation, the victim entry is going to be used in future*?
- How many entries do we need to make this probability low enough?

Birthday paradox

- Birthday paradox: m samples are drawn and put into bins 1, 2, ..., N with uniform probability. The probability of not having any conflict is at least
- $$e^{\frac{-m^2}{2N}}$$
- Collorary: $Q >= P^3$ then we are safe

Empirical study

- $P = 1\text{M}$
- $Q = 10 * \text{P}$ (90 bytes overhead for each page)
- $K = 10$