

Mome

Yvon Jégou

Summary

- Memory allocation

Mome.1

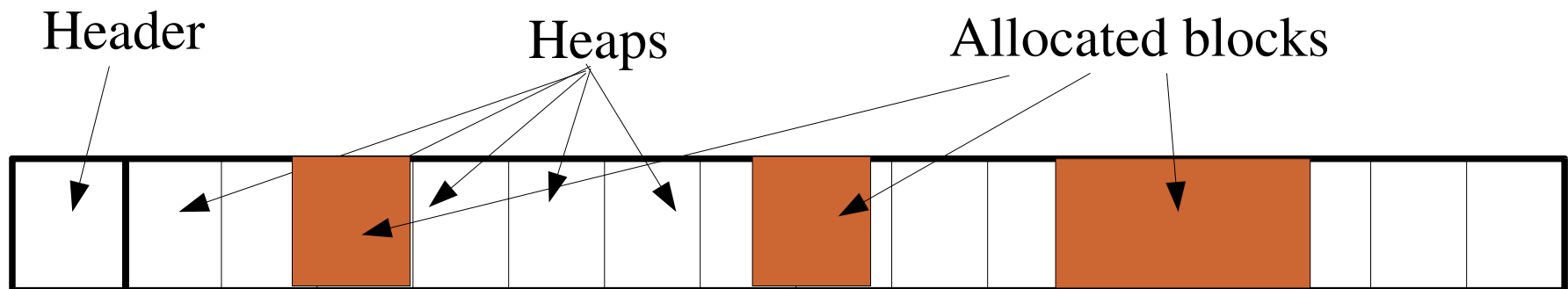
- Scalability
- Hierarchy
- Consistency models
- Cloning

Memory allocation

- implement a shared **malloc/free**
- symmetry:
 - allocate on some node
 - free on another
- efficiency:
 - avoid mutex locks
- load balance
 - all allocations by a single node?

MomeMalloc implementation

- create a large (~ 800 Mb) shared Mome segment
- mapped at the same address on all nodes
- simple memory global allocator protected by a distributed lock



MomeMalloc: heaps

- **address** -> **heapnum** (mask + shift on address)
- a heap can be
 - free: no allocation
 - shared: global allocation
 - dedicated to some node
- **owner[heapnum]** in global header
- heap size: 8 Mbytes

ptmalloc2

- **ptmalloc2** from Doug Lea / Wolfram Gloger
- each node: one or more **arenas**
- **arena**: set of heaps
- **malloc**: node allocates memory inside its arenas without global lock
- `free(ad)`: **owner[heapnum(ad)]**
 - free if local
 - global free if global (with lock)
 - send to owner if non local

Distributed lock

- on **malloc**: not enough space
 - take global lock
 - allocate a new heap (global allocation)
 - **owner[heapnum] = nodeid;**
- on **free**: if a heap is free, it can be returned to the global space
- page faults: during access to global header

Malloc and consistency

- MomeMalloc works with weak and relaxed consistency model.
- But: it is not possible to modify the consistency of a dynamically allocated region
- Future work...

Mome.1: scalability

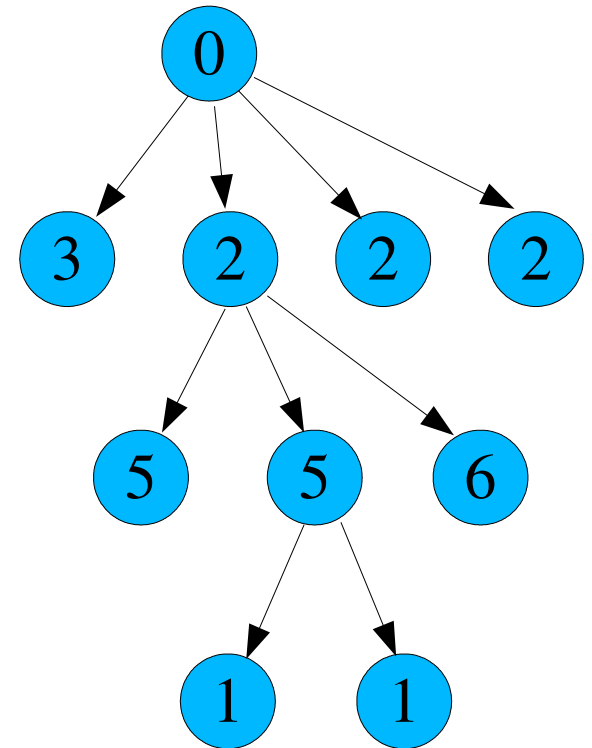
- We want more shared memory, more nodes
- 2^{32} pages, 2^{16} nodes
- dynamically add or remove nodes
- hierarchical implementation
- no global synchronization

Consistency models

- currently: strong consistency and weak consistency
- Mome.1:
 - release consistency
 - support for parallel reduction
- consistency model for a page: dynamically selected at the node and page level

Hierarchy implementation

- page manager id: $\text{pagenum} + \text{nodenum} + \text{tag}$
- tags
 - tag 0: root manager
 - tag 1: leaf SMP manager
 - others: intermediate managers
- management:
 - tags 0 and 1: Mome
 - others: applications



Cloning

- clone request from a process:
 - page descriptor points to clone descriptor until it is modified (share data)
 - the clone-page is read-only
 - the clone-page can be mapped in memory
- asynchronous implementation
- no page move during cloning (except in the multiple-writers case)

Checkpointing

- Checkpointing of a region:
 - clone all pages of the region (in parallel)
 - add a reference to each clone-page
 - resume the application(s)
 - checkpoint the clone-pages in the background (duplicate the clone-pages as in Mome.0.8)
- Expect fast checkpointing