

Mome, Checkpointing

Yvon Jégou
Paris Project,
IRISA/INRIA
FRANCE

Overview

- Mome memory model
- Mome page management
- Checkpointing
- Current work

Mome memory model

- Posix memory model

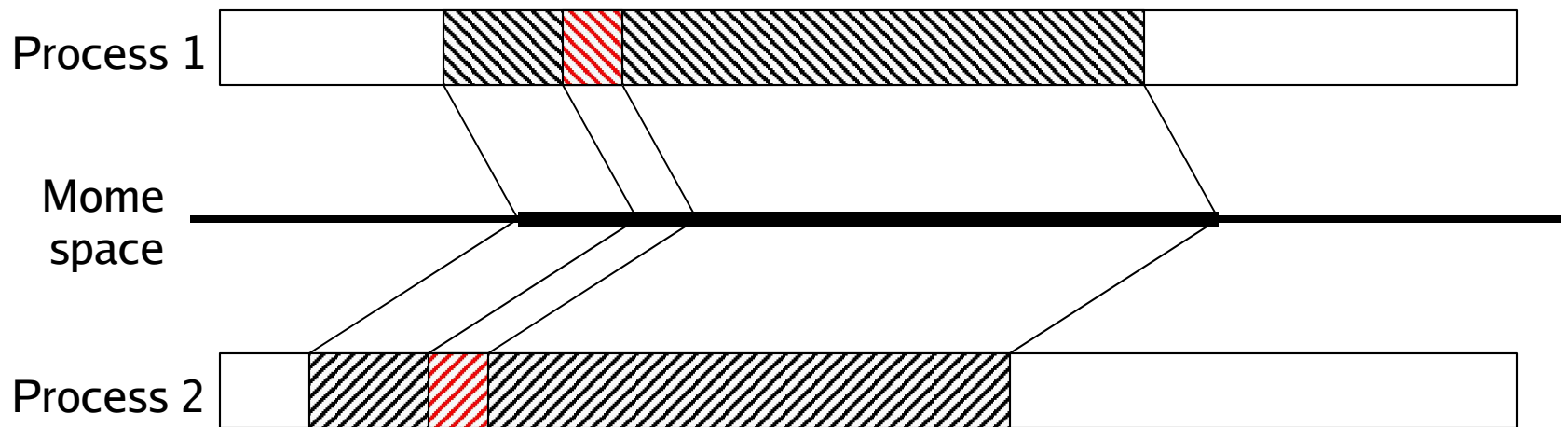
- Posix **mmap** → **MomeMMap**

single node

multiple nodes



Mome memory model



F77 common block sharing

```
double precision tab(n,m)
common /arrays/tab
..
call MomeMMap(tab(1,1), sz(tab), WRITE,
              FIXED, seg1, 0)
..
tab(i, j)= ...
...= tab(k, l)...
```

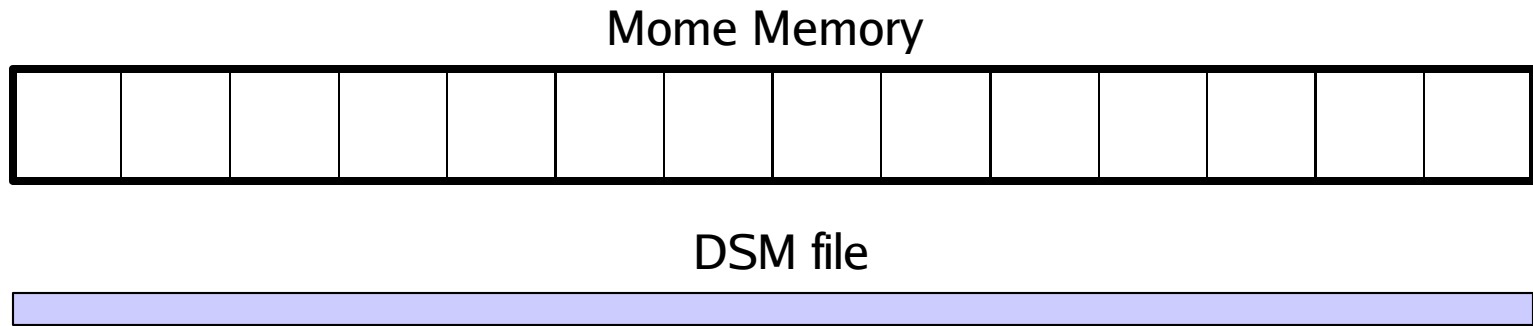
Mome page management

- Directory-based
 - Directory records status of page in each node
 - No page home
- (asynchronous) directory migration or redistribution

Multiple mappings

- The same page can be mapped multiple times on the same node
 - By the same process (*synchronous*)
 - By different processes of the same node (*asynchronous*): persistent data repository
 - By a distant process (no shared memory)
 - Hierarchical DSM (*work in progress*)

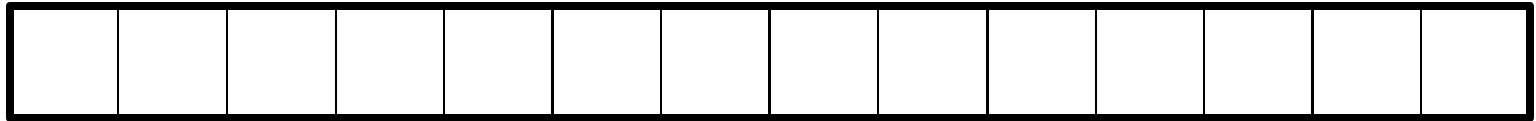
Internal node page management



- On each node: Mome memory is mapped on the DSM file (temp file)
- Application pages made accessible through aliasing on Mome memory (temp file **mmap**)

Internal node page management

Mome Memory



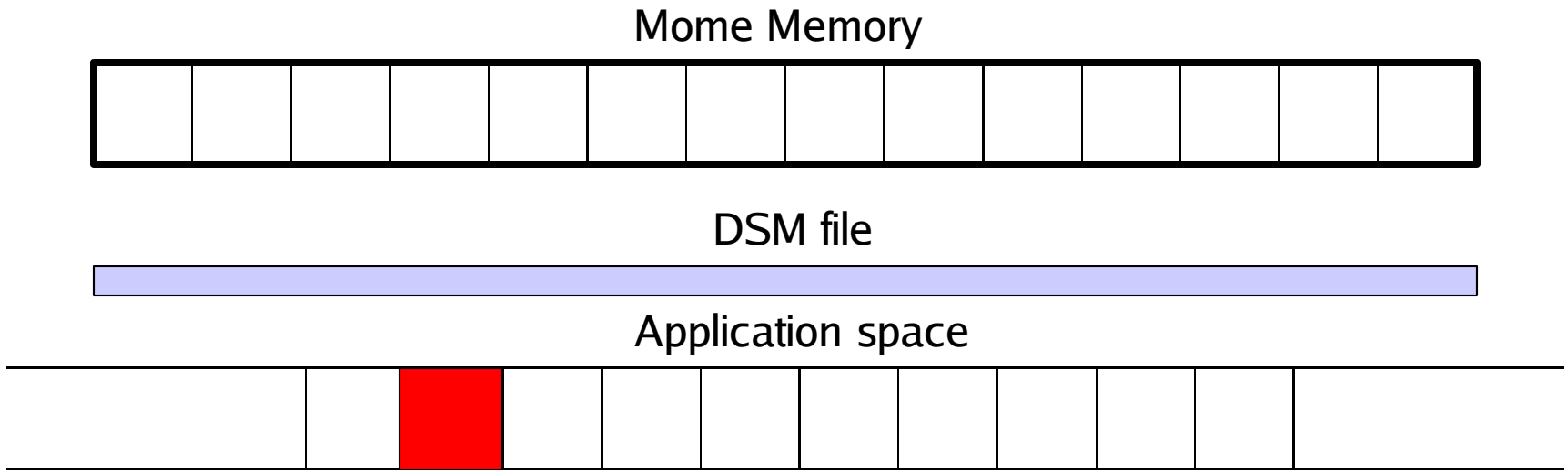
DSM file



Application space

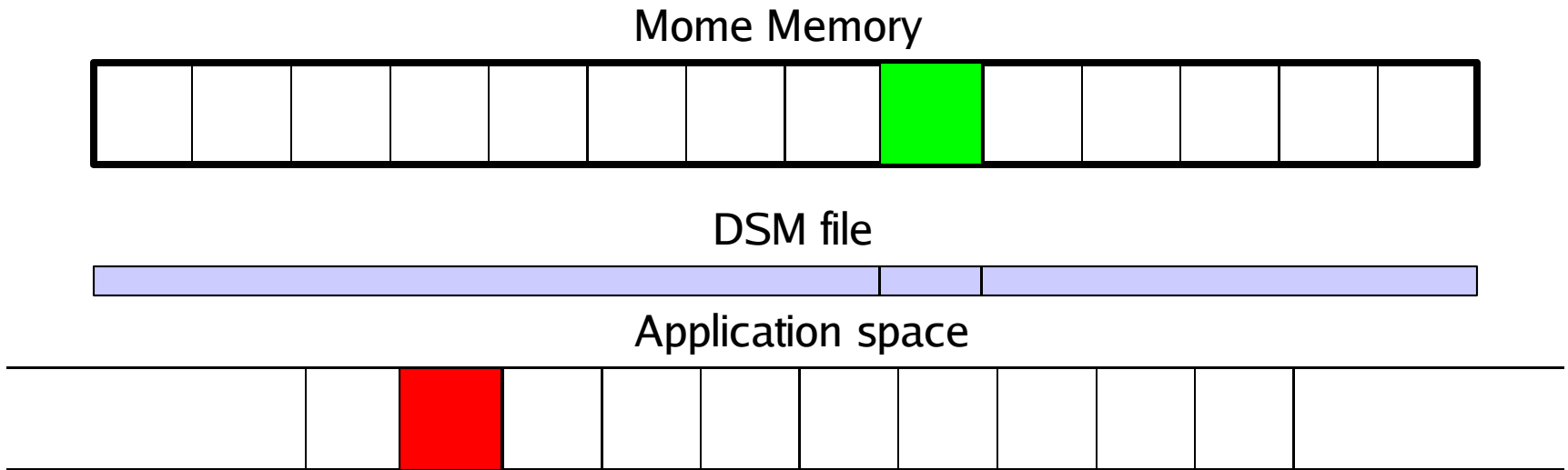


Internal node page management



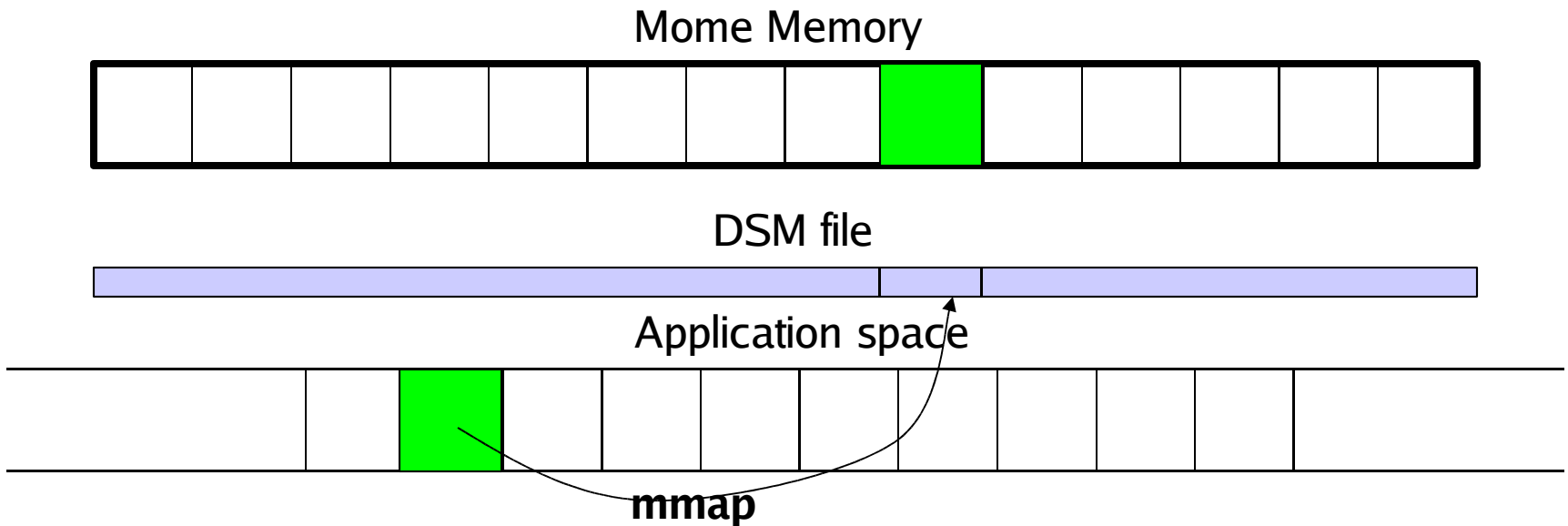
- Page fault: page made available in Mome memory

Internal node page management



- Page fault: application page is aliased with page in Mome memory

Internal node page management



- Application process can read/write the page

DSM memory management

- Each page: reference counter
 - Mapping references
 - Communication layer references
 - Copy-on-write (*refcount 1*)
- Free page list
- DSM memory management thread
 - Free unused pages
 - Distant page-out (*swapping*)

Checkpointing

- Page checkpoint on some node
 - Increment the *refcount* of a page
 - The copy of the page is locked in the node
- Checkpoint on a node
 - List of page references
- Global checkpoint
 - Each page checkpointed on two nodes (*at least*)

Checkpointing in Mome.0.8

- Global synchronization (barrier)
- Each node: list of present/seen page
- global/distributed merge of lists
- Select two nodes in charge of each page
- Each node: list of pages to checkpoint
- For each page: request its presence
 - Increment refcount
- Commit (barrier)
- Destroy previous checkpoint (*decrement refcounts*)

Checkpointing in Mome.0.8

- Checkpoint the DSM only
- Does not checkpoint the application

Restart after failure

- Mome initialization
- Each node: analyze Mome memory and checkpoints from Mome file (/tmp/file)
- Select coherent state (stamps)
- Each node: creates list of recoverable pages
- Global/distributed list merge
 - Check that a copy of each page is present
- Each node: a list a pages to recover

Performance analysis

- Need for global synchronization
 - Multiple applications ?
- Application waits until commit before resuming execution
 - Page presence analysis
 - Page duplication
- After checkpoint: first write needs to invalidate copies

Current work: Mome.1

- Add release consistency model (for OpenMP)
- Hierarchical DSM for “cluster of clusters”
- Support for large number of nodes
- Support for large number of pages 2^{32}
- Support for page *snapshot*

Checkpointing in Mome.1

- Page snapshot
 - Page snapshot: transaction with page manager: receive a handle to a twin of the page
 - No page move for snapshot (*except for multiple-writer case: merge*)
- Application can restart once all handles have been received
- Snapshot of pages can be checkpointed in the background (in parallel with application)
- Expected performance:
 - No wait for duplication,
 - No page invalidation on first write
 - Checkpointing nodes independant from application nodes

