Today

Lesson 8 Videos

⇒ — LRVM

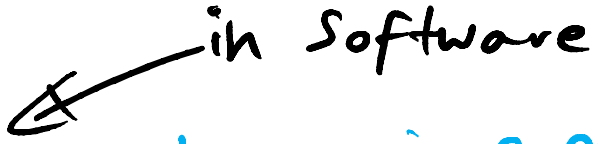
Friday

— RioVista

— Hg

(* please watch videos *)

LRVM



- persistent memory layer in support of system services

Why ?

Persistence

Why?

- need of OS subsystems

How?

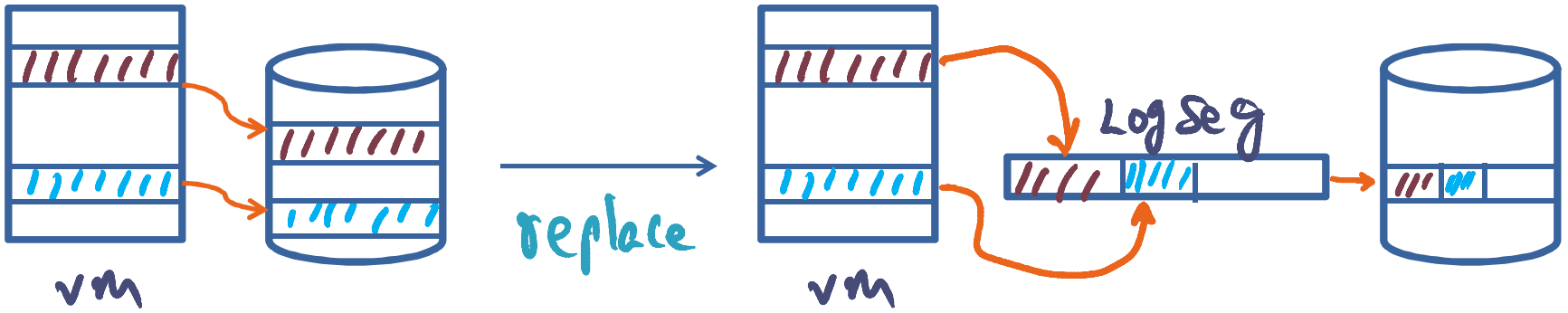
- make virtual memory persistent

Who will use it?

- Subsystem designers if it is performant

How to make it efficient?

- Use persistent logs to record changes to VM



Server Design

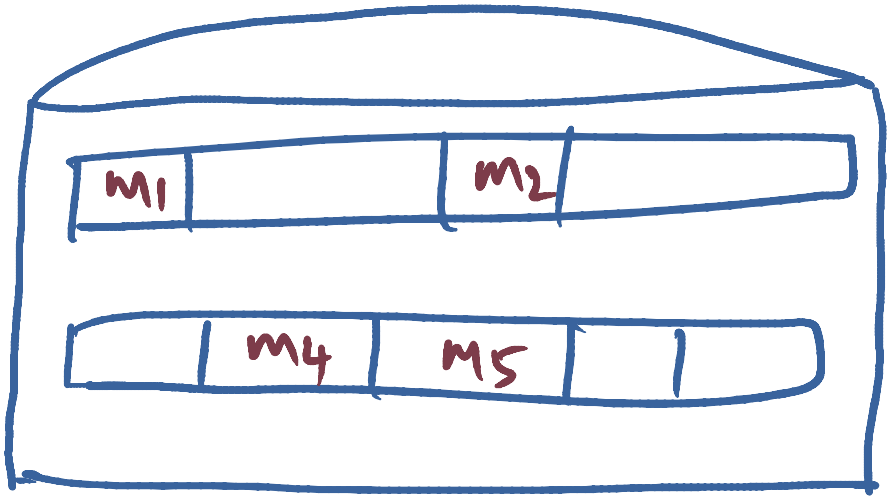
- persistent metadata $m_1, m_2, \ldots, m_n$
- normal data structures + code →

Virtual address space of server

⇓

Create external data segments to back persistent data structures

- apps manage their persistence needs



seg1

seg2

Designer's choice to use single or multiple data segments

RVM primitives

Initialization

- initialize(options)
- map(region, options)
- unmap(region)

RVM primitives

Initialization

- initialize(options)
- map(region, options)
- unmap(region)

Body of server code

- begin_xact(tid, restore_mode)
- set_range(tid, addr, size)
- end_xact(tid, commit_mode)
- abort_xact(tid)

RVM primitives

Initialization

- initialize(options)
- map (region, options)
- unmap(region)

GC to reduce log space

- flush ( )
- truncate ( )

Body of server code

- begin_xact (tid, restore_mode)
- set_range(tid, addr, size)
- end_xact (tid, commit_mode)
- abort_xact (tid)

RVM primitives

Initialization

- initialize(options)
- map(region, options)
- unmap(region)

GC to reduce log space

- flush()
- truncate()

} done by LRVM automatically

⇓

Provided for app flexibility

Body of server code

- begin_xact(tid, restore_mode)
- set_range(tid, addr, size)
- end_xact(tid, commit_mode)
- abort_xact(tid)

RVM primitives

Initialization

- initialize(options)
- map(region, options)
- unmap(region)

GC to reduce log space

- flush()
- truncate()

} done by LRVM automatically

⇓

Provided for app flexibility

Body of server code

- begin_xact(tid, restore_mode)
- set_range(tid, addr, size)
- end_xact(tid, commit_mode)
- abort_xact(tid)

Miscellaneous

- query_options(region)
- set_options(options)
- create_log(options, len, mode)

RVM primitives

Initialization

- initialize(options)
- map(region, options)
- unmap(region)

Body of server code

- begin_xact(tid, restore_mode)
- set_range(tid, addr, size)
- end_xact(tid, commit_mode)
- abort_xact(tid)

GC to reduce log space

- flush()
- truncate()

} done by LRVM automatically

⇓

Provided for app flexibility

Miscellaneous

- query_options(region)
- set_options(options)
- create_log(options, len, mode)

⇒ Simplicity - small set of primitives

How the server uses the primitives

```
Initialize address space from Ext segs
        ⋮
begin_xact(tid, mode);
    set range (tid, base_addr, #bytes);
        ⋮
    write metadata m1          ;// contained in range
        ⋮
    write metadata m2          ; // contained in range
end_xact (tid, mode);          //or this can be abort
```

How the server uses the primitives

```
Initialize address space from Ext segs
        ⋮
begin_xact(tid, mode);
    set range (tid, base_addr, #bytes);
        ⋮
    write metadata m1
        ⋮
    write metadata m2
end_xact (tid, mode);
```

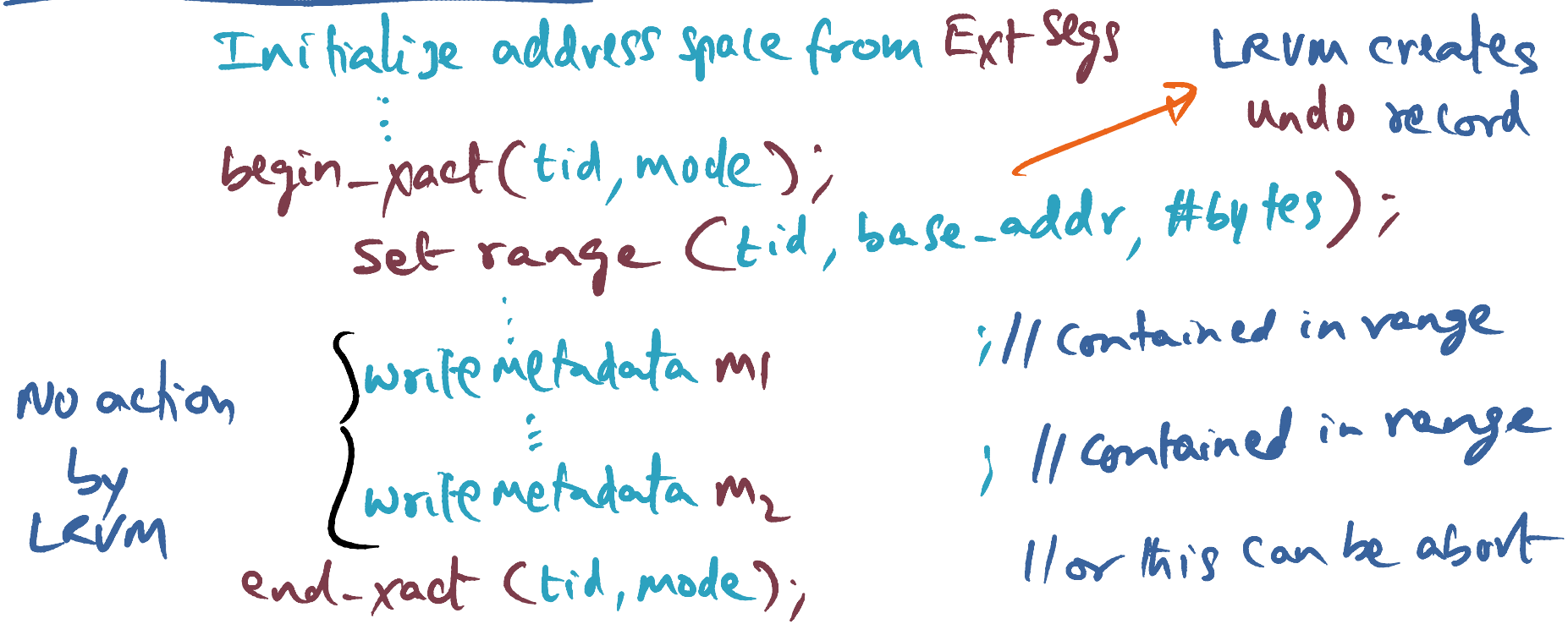
LRVM creates undo record

; // Contained in range

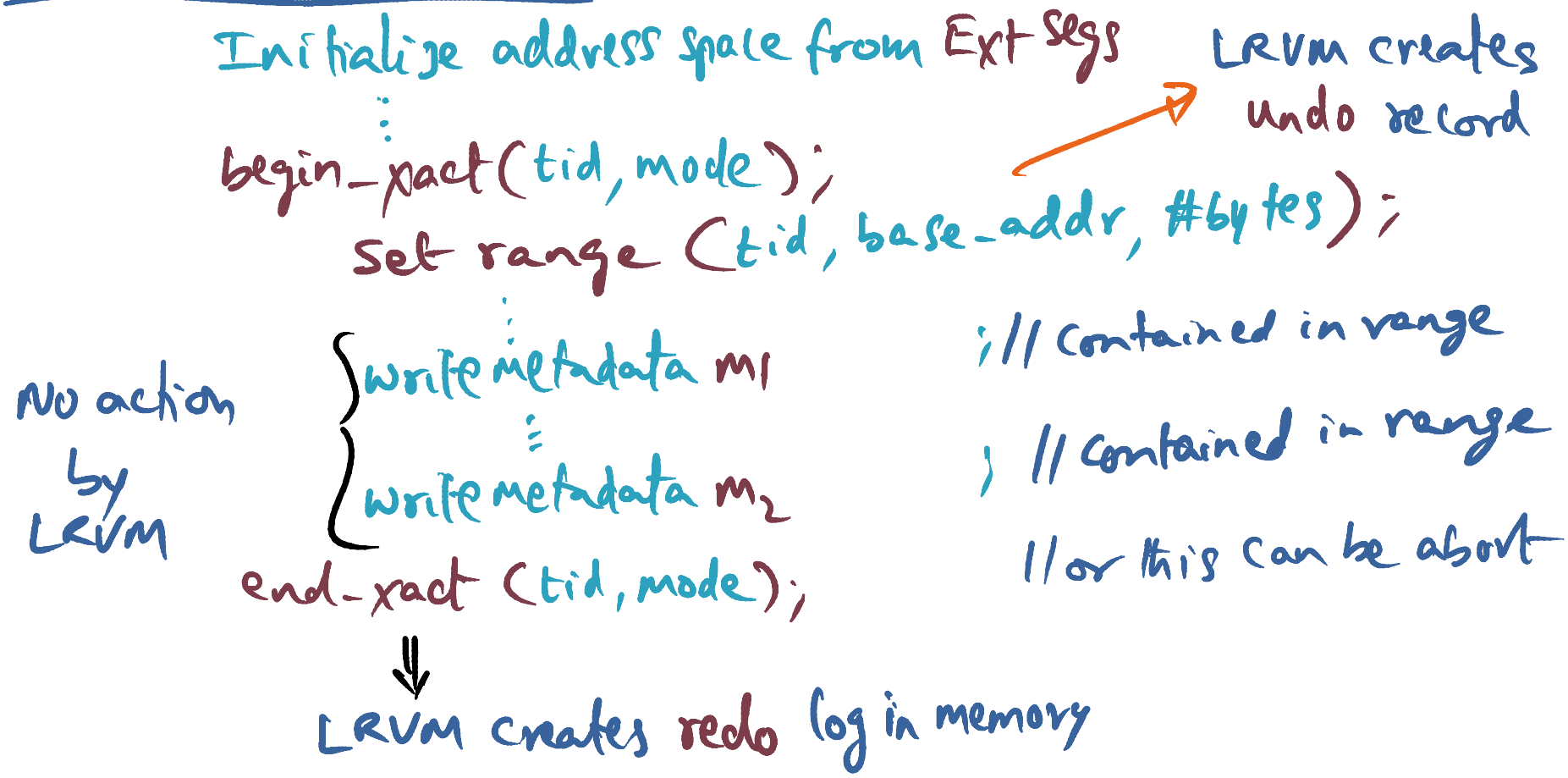
; // Contained in range

// or this can be abort

How the server uses the primitives



How the server uses the primitives



How the server uses the primitives

Initialize address space from Ext segs

```
begin_xact(tid, mode);
    set range (tid, base_addr, #bytes);   // Contained in range
        :
        write metadata m1
        :
        write metadata m2                 // Contained in range
end_xact (tid, mode);                     // or this can be abort
```

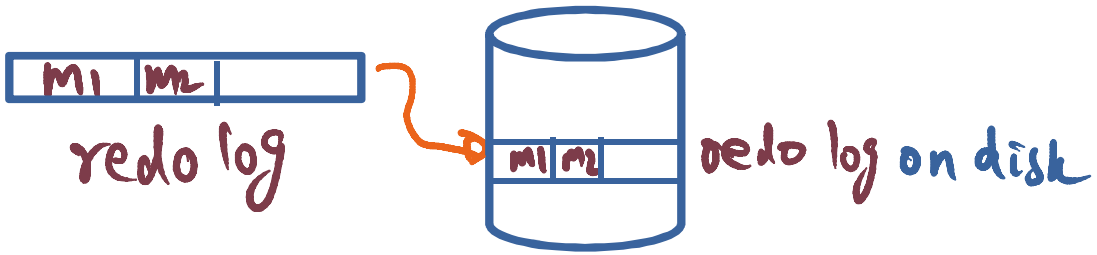
LRVM creates undo record

No action by LRVM

⇓

LRVM creates redo log in memory

– flush to disk sync or later depending on mode



How the server uses the primitives

Initialize address space from Ext segs

```
begin_xact(tid, mode);
    set range (tid, base_addr, #bytes);
        ;// Contained in range
    write metadata m1
        :
    write metadata m2
        ; // Contained in range
end_xact (tid, mode);
    // or this can be abort
```

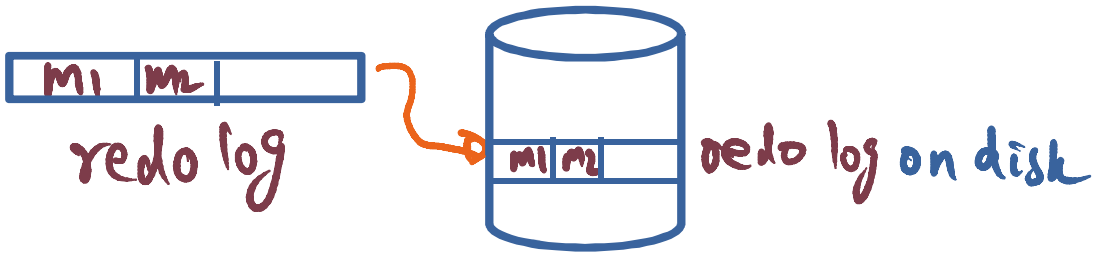
LRVM creates undo record

No action by LRVM

LRVM gets rid of undo record

LRVM creates redo log in memory

- flush to disk sync or later depending on mode



How the server uses the primitives

Initialize address space from Ext segs

```
begin_xact(tid, mode);
set range (tid, base_addr, #bytes);   // Contained in range
    write metadata m1
    :
    write metadata m2                 // Contained in range
end_xact (tid, mode);                 // or this can be abort
```

LRVM creates undo record

No action by LRVM

LRVM gets rid of undo record

restore from undo record

LRVM creates redo log in memory

- flush to disk sync or later depending on mode