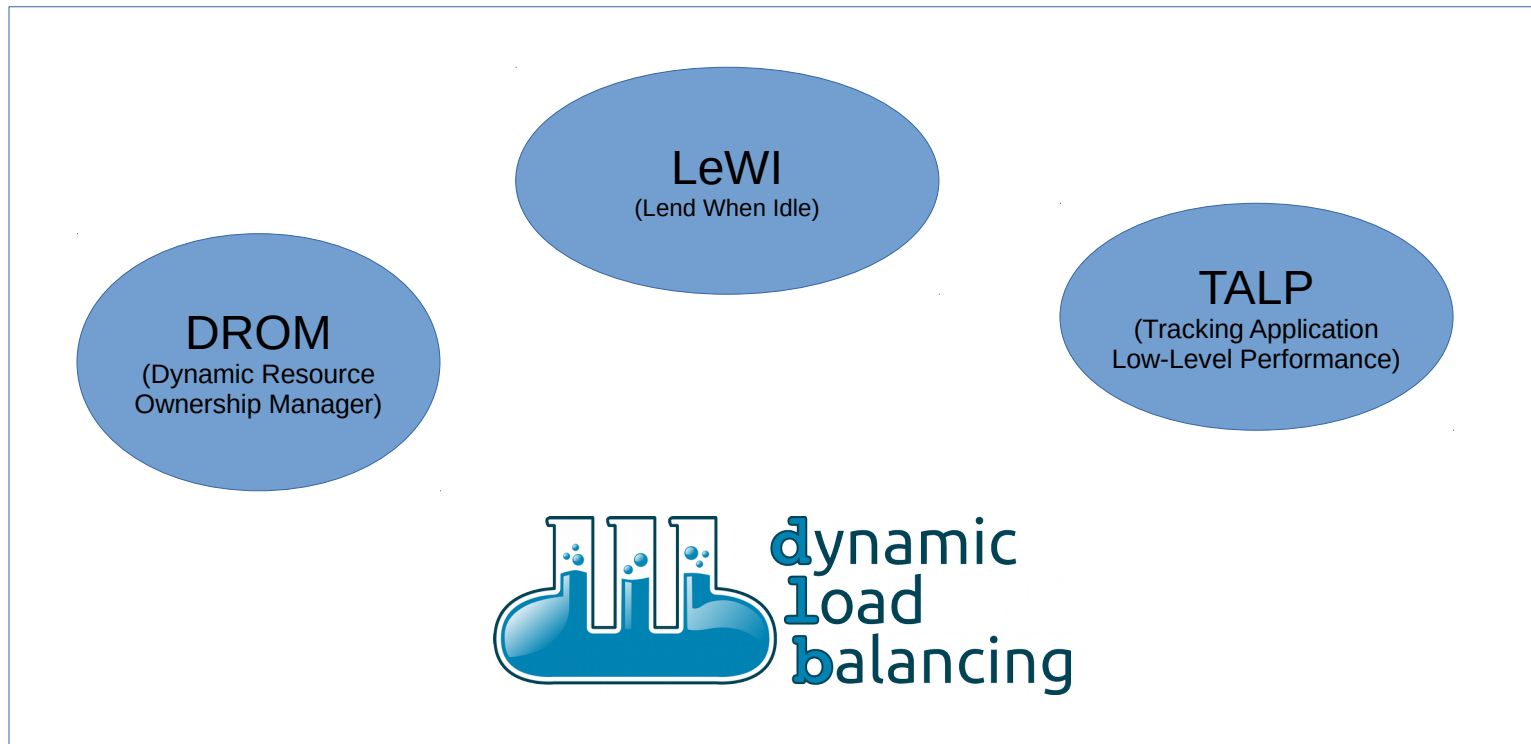# DLB development guide

Abstract View

➢ DLB Modules

    – Each one enabled with DLB_ARGS="--lewi --drom --talp"

LeWI features

> CPU attributes:
> - state: IDLE | BUSY | DISABLED
> - guest: <pid> | NOBODY
> - owner: <pid> | NOBODY

> Lend: state:=IDLE and find new guest

> Reclaim: state :=BUSY and set guest only if NOBODY (only for owned CPUs)

> Acquire: same as Reclaim, but without limitation

> Borrow: guest:=<pid> if guest == NOBODY

> Return: guest:=owner if state == BUSY

DROM features

- Process Attributes:
  - current_process_mask
  - future_process_mask
  - stolen_mask

- Set process mask (admin process):
  - future[i] := <new_mask>
  - $\forall_{p\ if\ p!=i}$, future[p] := future[p] – future[i]
  - $\forall_{p\ if\ p!=i}$, stolen[p] := stolen[p] ∪ (current[p] – future[i])

- Apply new process mask (application process):
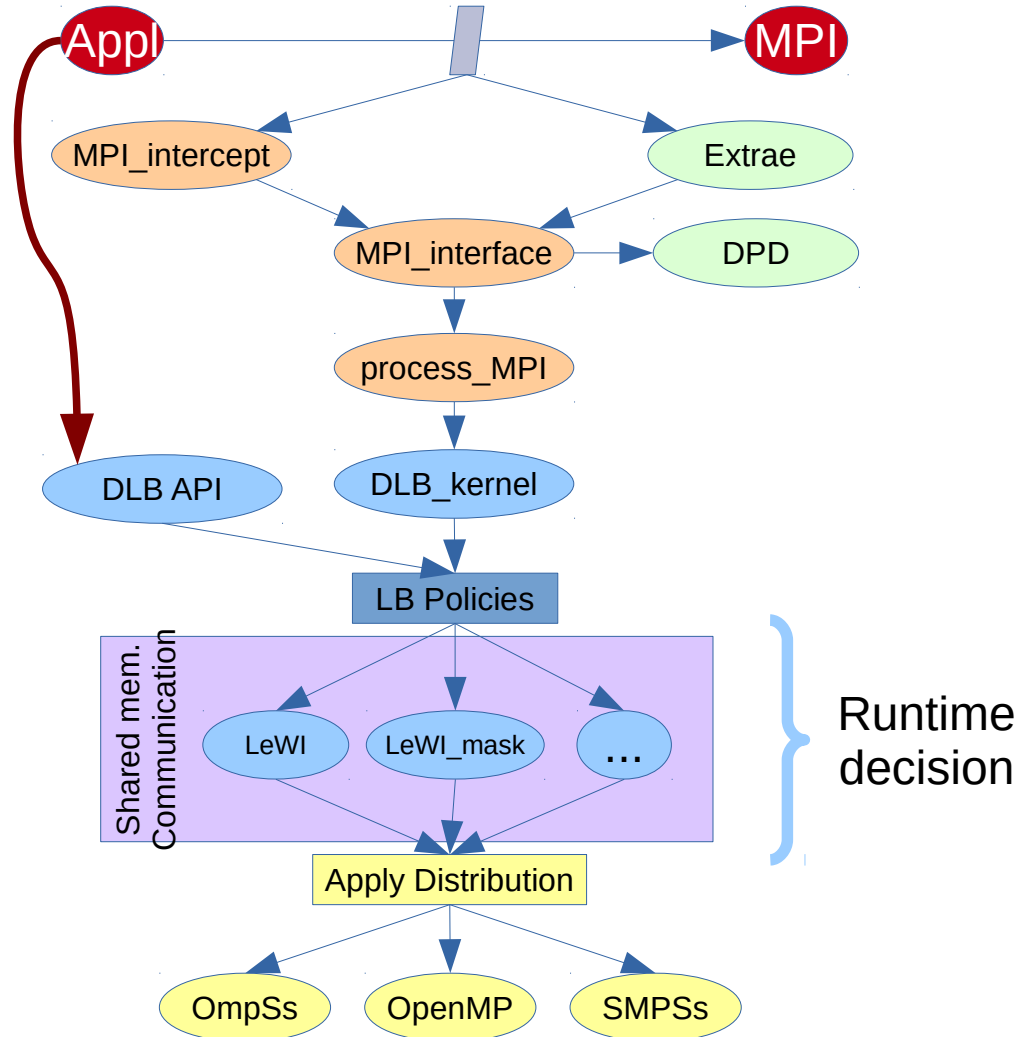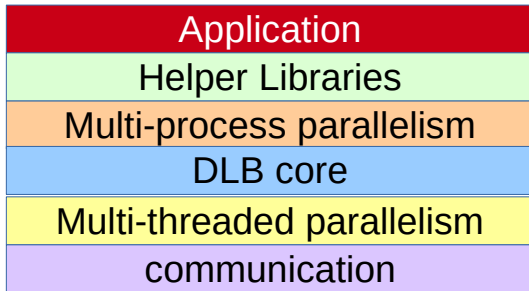 - current[i] := future[i]

TALP features

➢ Collect performance metrics

 – CPU state profiling

 – Useful CPU time per process

➢ In the future, apply knowledge to DLB scheduling

Software Layers

Shared Memory

➢ List of shared memories:

- – "cpuinfo": CPU indexed shared memory
- – "procinfo": Process indexed shared memory
- – "lewi": minimalistic shared memory for non-mask support
- – "async": asynchronous message queue shared memory
- – "barrier": prototype for custom barrier between processes

Barcelona Supercomputing Center
Centro Nacional de Supercomputación
BSC

Shared Memory: cpuinfo

➢ node_info, indexed by cpuid:

- id's

- lewi info: state, owner, guest

- drom info: owner

- queue for processes requesting the CPU

- statistics

  - Accumulated time or percentage of each state

  - …

Shared Memory: procinfo

➢ process_info, indexed by subprocess id:

- id's

- drom info: current, future and stolen masks

- Statistics (per process):

  - cpu_usage
    - Reading granularity? (avg, since last read, …)

  - load_avg
    - Readings of cpu_usage for the lasts 1, 5, 15 mins

  - …

Programming Model management

➢ TBD

Coding guidelines

➢ No tabs, 4 space indentation

➢ Curly braces block begins in the same line

➢ Avoid trailing spaces

➢ Avoid global variables, use static if possible to reduce scope

➢ Avoid literals, prefer anonymous enums or macros