東京 2025

LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025

Cache Aware Scheduling

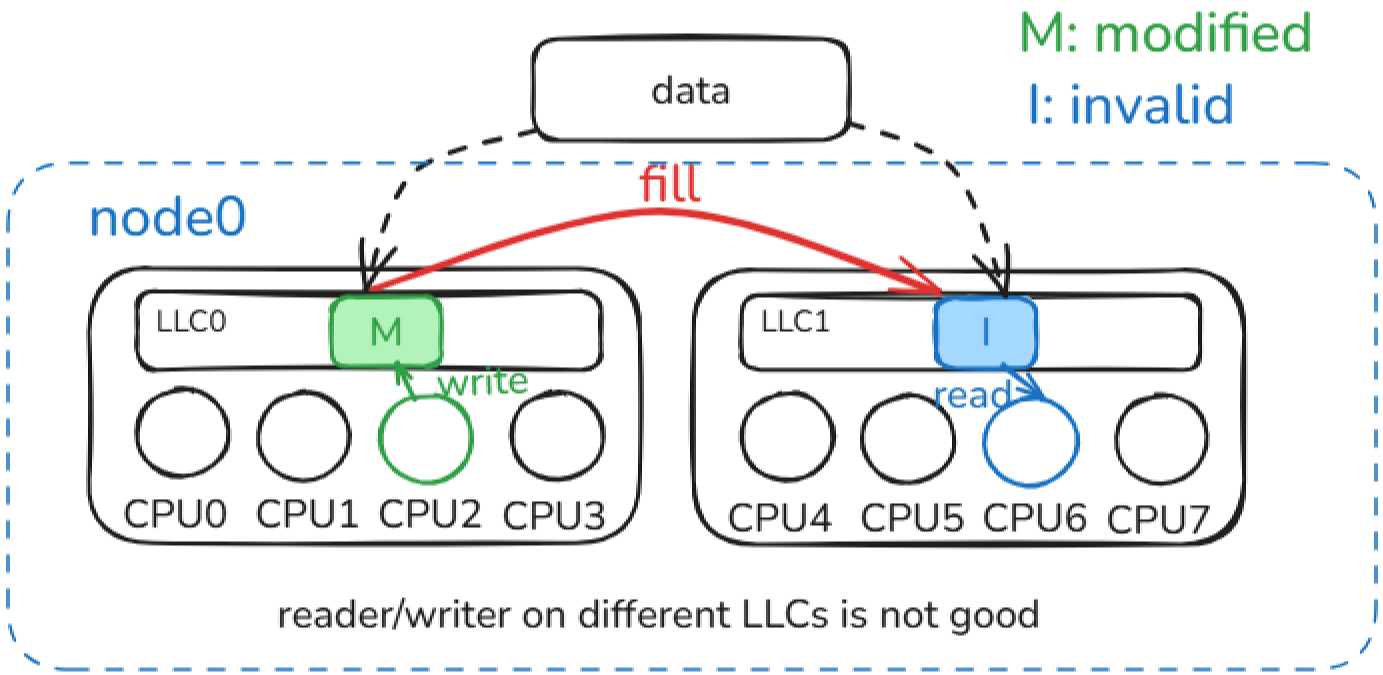
Tim Chen,
Chen Yu

東京 2025
LINUX PLUMBERS CONFERENCE
TOKYO, JAPAN / DEC. 11-13, 2025

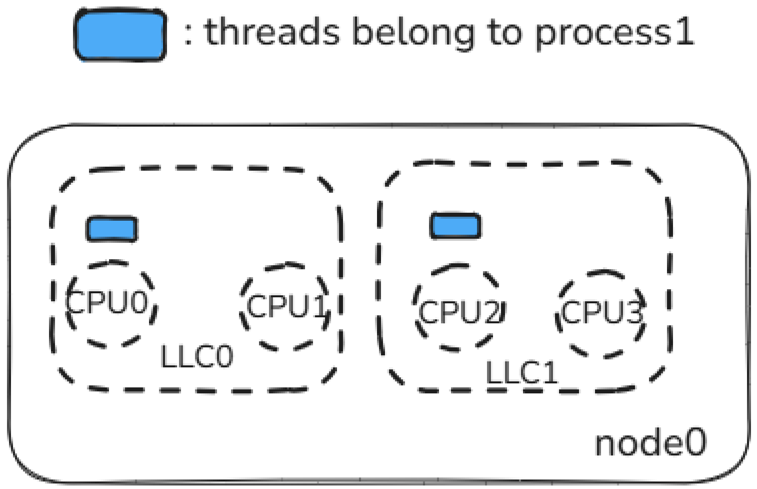
Outline

- Problem Statement
- Proposal and current status
- Seek feedbacks on tasks aggregation criteria

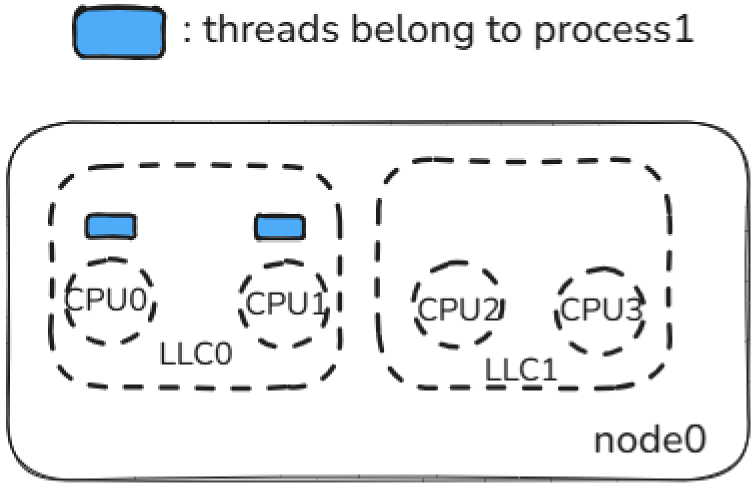
Problem statement: cross LLC access penalty



Problem statement: current load balancer

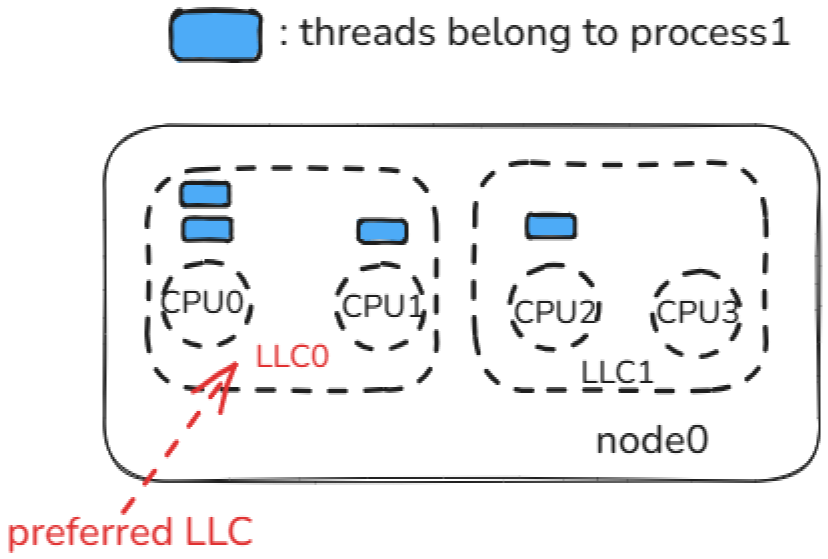


Proposal: expected behavior of load balancer



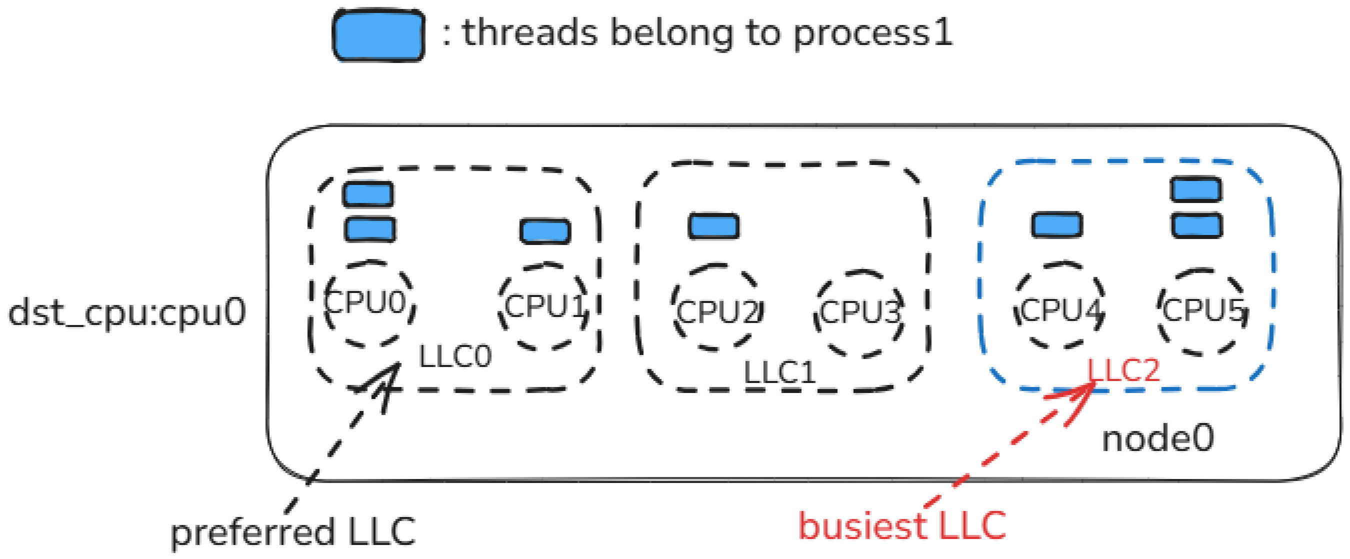
Proposal: let load balance aggregate the tasks

- Step1: calculate the preferred LLC for the process(tick)



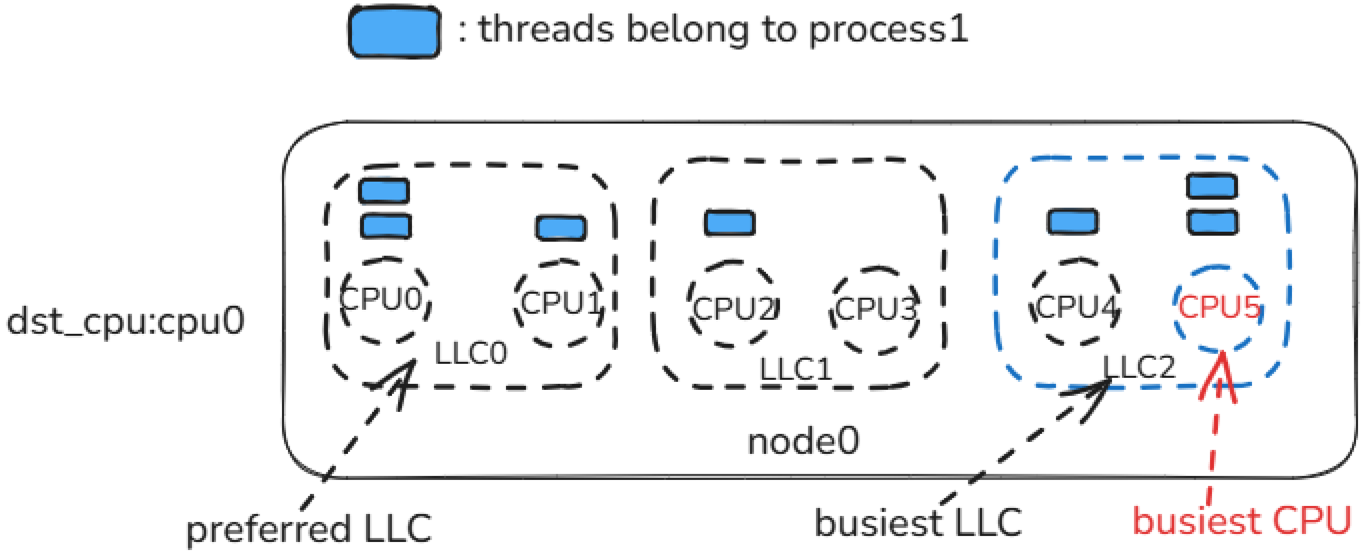
Proposal: let load balance aggregate the tasks

- Step2: find the busiest source LLC during load balance



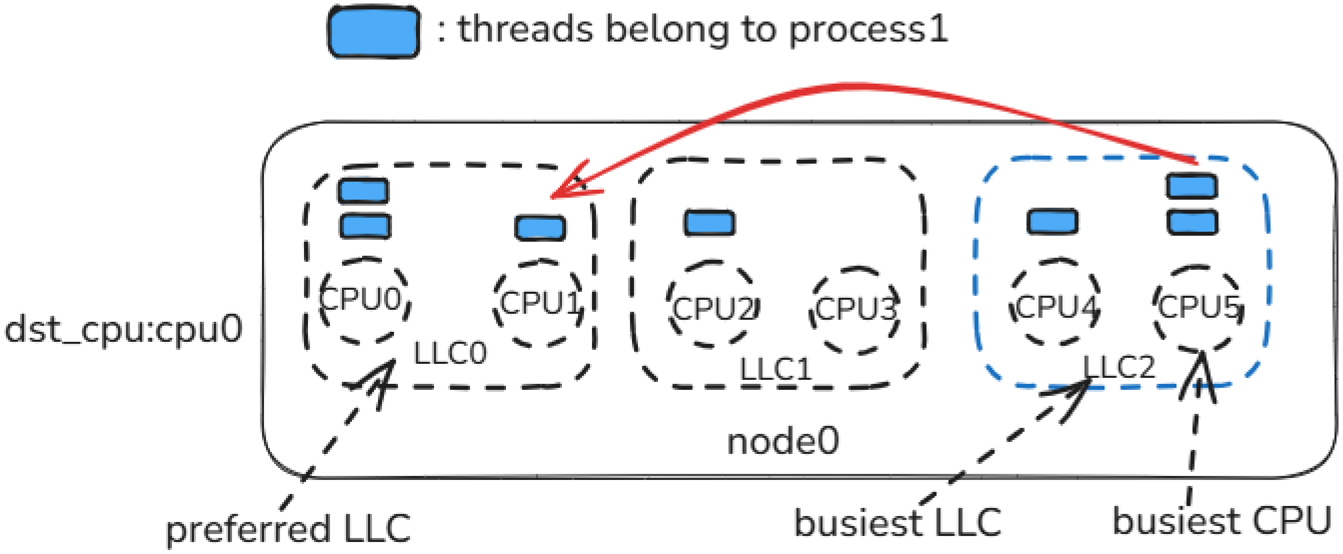
Proposal: let load balance aggregate the tasks

- Step3: find the busiest source CPU



Proposal: let load balance aggregate the tasks

- Step4: sort and migrate the threads



Benchmark results

Xeon, 2 sockets, 60 cores/socket, DRAM interleaving, 2LLCs/Node, turbo off, CPUfreq performance, deep C-states disabled	
benchmark	sched_cache vs baseline improvement
hackbench pipe	30% (nr_running < llc_sd_size)
hackbench socket	15% (nr_running < llc_sd_size)
RISC-V Xiangshan Simulator Chacha20 encryption	26%
schbench 99.0th wakeup latency	[7%, 35%] (1 messager)
Others	1 case 8% regression due to over-aggregation
EPYC Turin, Phoronix reported improvements in Ethr,DaCapo, Renaissance,ClickHouse,Apache IoTDB,Memcached, PostgreSQL, etc.	

How to do fine-gained control?

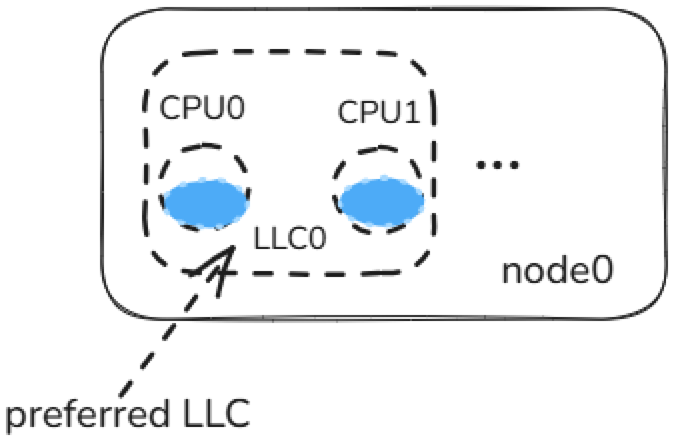
- Can we use prctl to enable/disable aggregate tasks on a per process basis?

- Can we group tasks to aggregate from a per-process basis to per cgroup/numa_group?

When to do task aggregation?

- condition1

🟦 : threads belong to process0



util_pref_llc < 50%

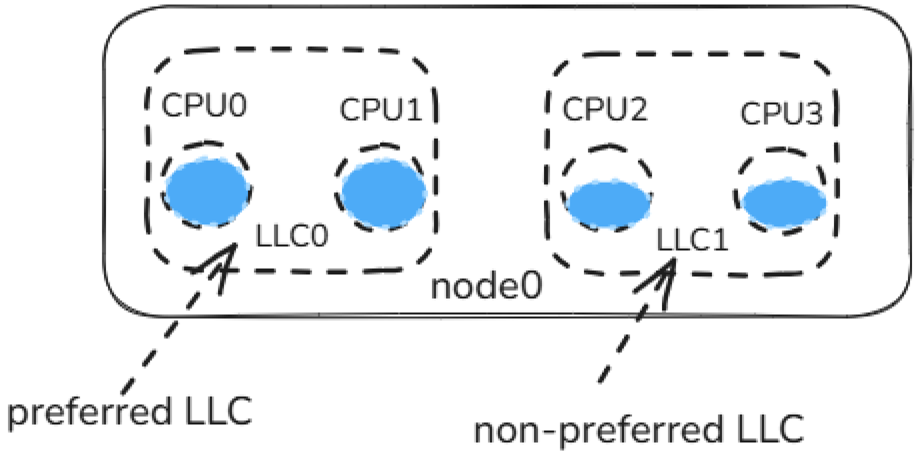
move task to preferred LLC

debugfs tunable

When to do task aggregation?

- condition2

🟦 : threads belong to process0

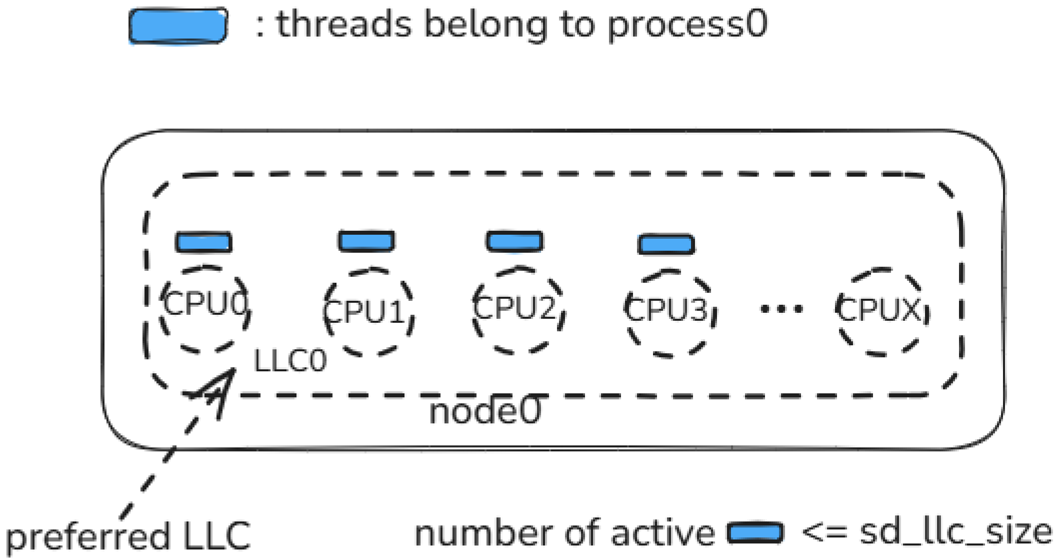


util_pref_llc - util_non_pref
< 20%
move task to preferred LLC

debugfs tunable

When to do task aggregation?

- condition3



debugfs tunable

When to do task aggregation?

- condition4 ：
  number of physical pages in used <= LLC cache size

  debugfs tunable

Thank you!

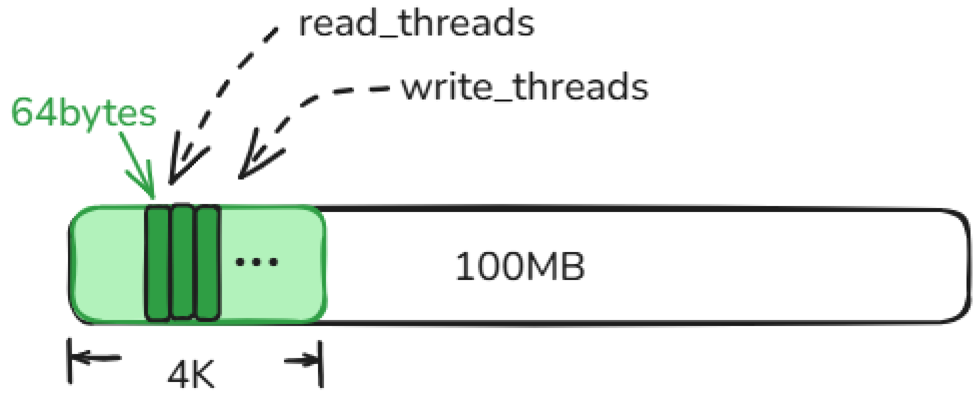
東京 2025

LINUX PLUMBERS CONFERENCE

TOKYO, JAPAN / DECEMBER 11-13, 2025

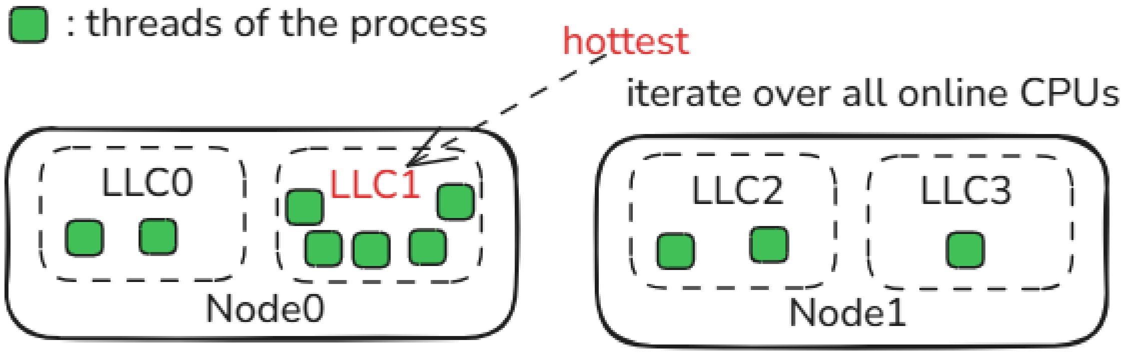
Appendix

Problem statement: simple cache contention test

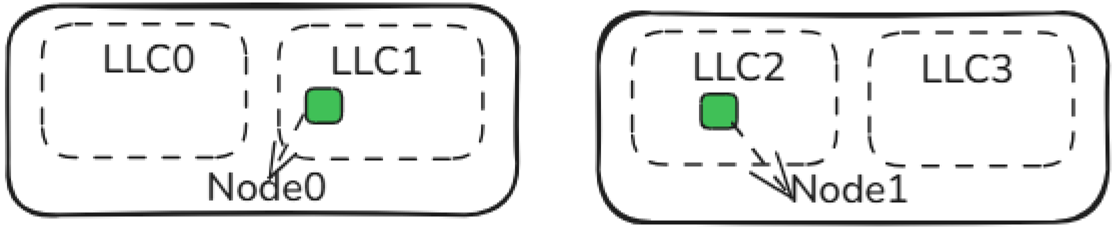


- Len Brown's benchmark shows up to **36%** difference between: bind to one LLC vs free run

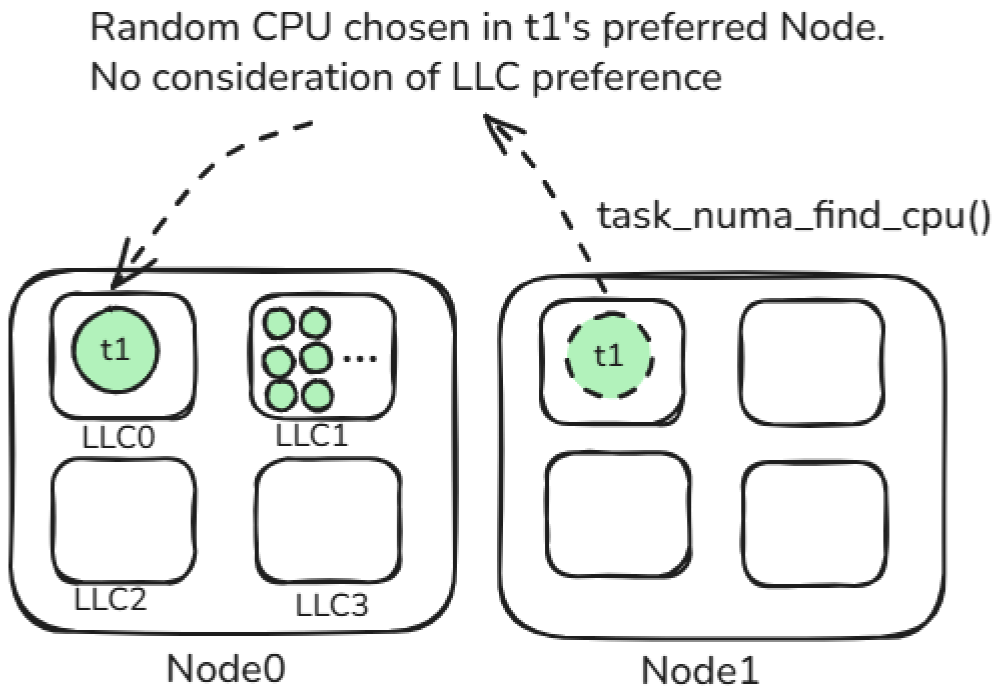
Seek feedback: how to reduce the cost of CPUs scan?



only scan within preferred node? --> whose preferred node?



Problem statement: Defficiency of NUMA load balancer


Random CPU chosen in t1's preferred Node.
No consideration of LLC preference
task_numa_find_cpu()
t1
...
LLC0
LLC1
LLC2
LLC3
t1
Node0
Node1

Links

- Latest version: https://lore.kernel.org/all/cover.1764801860.git.tim.c.chen@linux.intel.com/