Rework load_balance

Vincent Guittot
Linux Plumber Conference
Scheduler MC
9 September 2019

Status

- Patchset on LKML
  - v2 https://lkml.org/lkml/2019/8/1/529
  - v3 still under preparation

- Changes:
  - Extend groups classification
  - Define different type of migration
  - Better take into account new metrics & remove old heuristics
  - Fix some suboptimal tasks placements
  - Use load_avg instead of runnable_load_avg

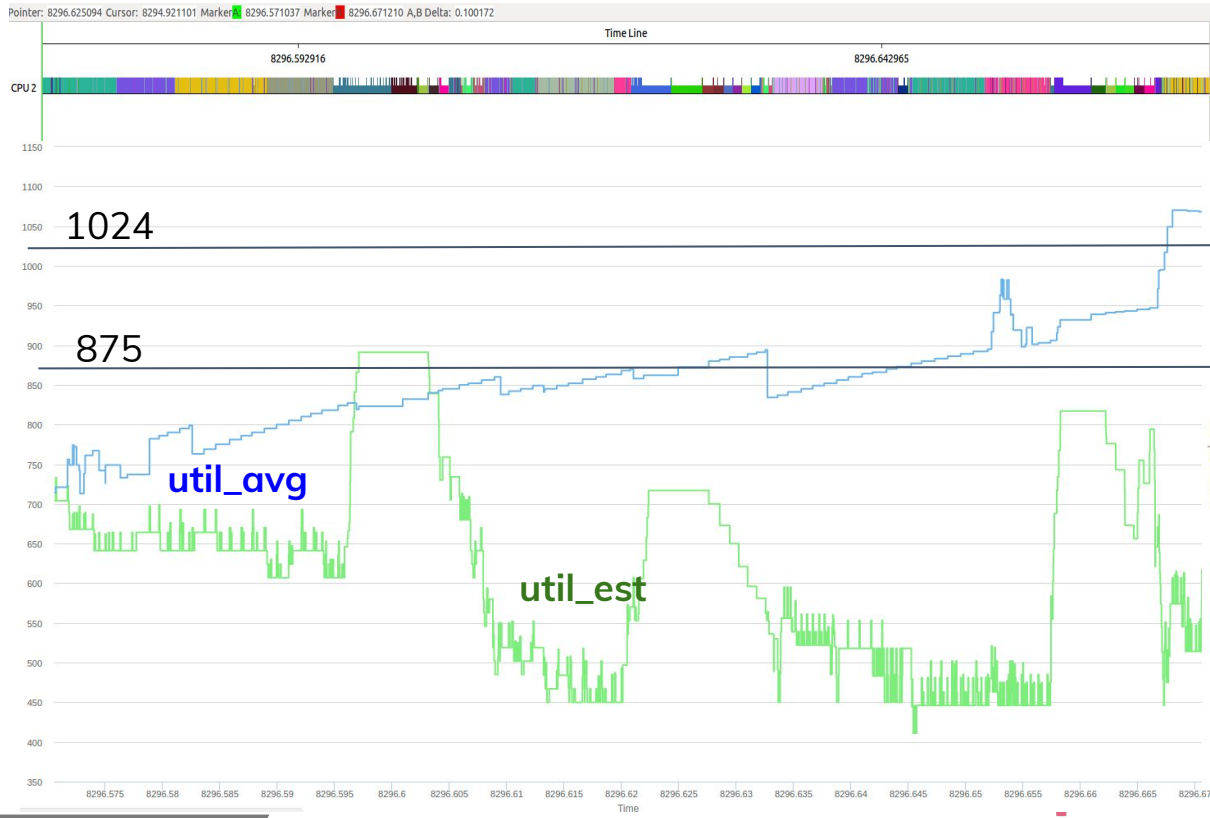
- Among fixed UCs:
  - 1 task per CPU
  - Better spread tasks in cgroup on numa
  - Preemption by other classes

Open item #1 runnable load vs load

- Runnable load introduced to
  - Fix cases with huge blocked load
- Load balance
  - Replaced runnable_load_avg by load_avg
- Wakeup path uses runnable load
  - Align policy with load_balance
  - A proposal submitted to use on load_avg but need refinements
- Numa stats uses runnable_load too
  - Study the impact of aligning with load_balance()

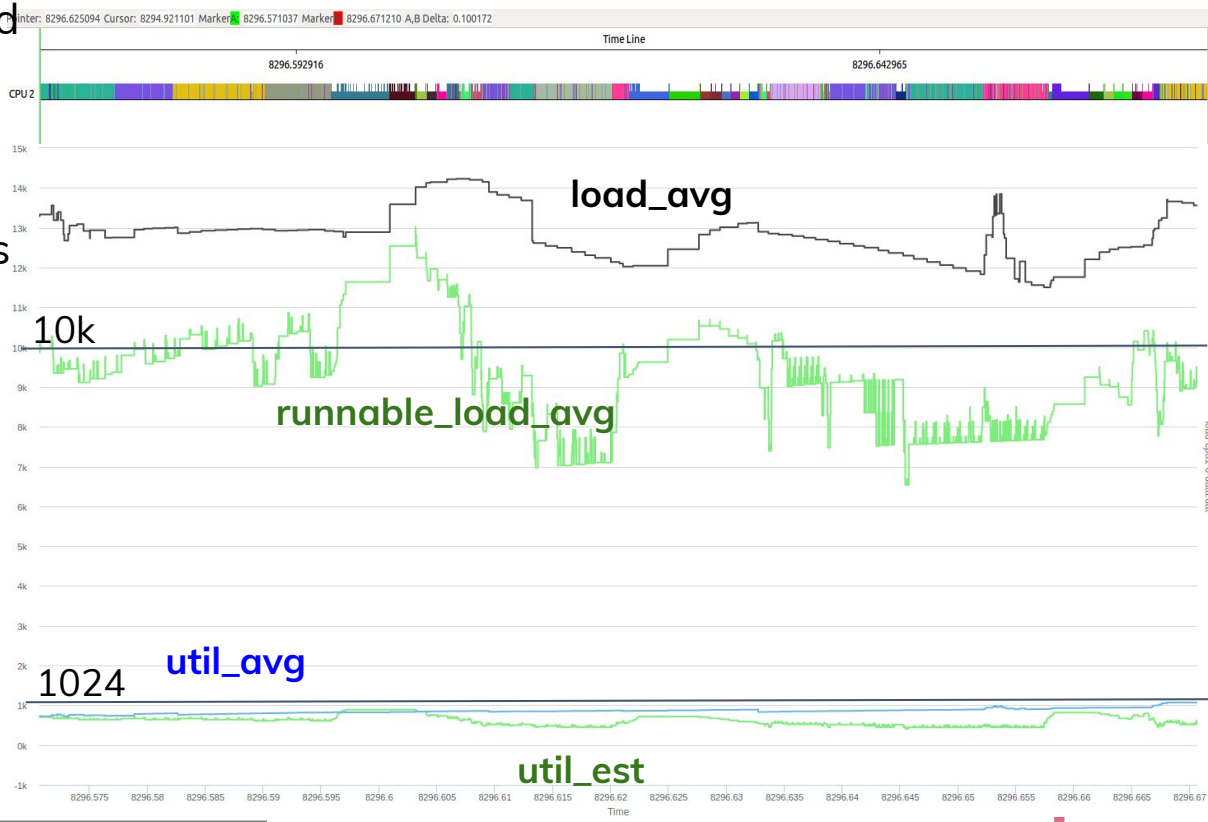
Open item #2 : detection of overloaded state

- How to better detect overloaded CPU/groups ?
- util_avg/util_est can be temporarily low after migration



Open item #2 : detection of overloaded state

- But load and runnable load stay quite high
  - tasks @ root with nice 0
- Nice weight screws up this information



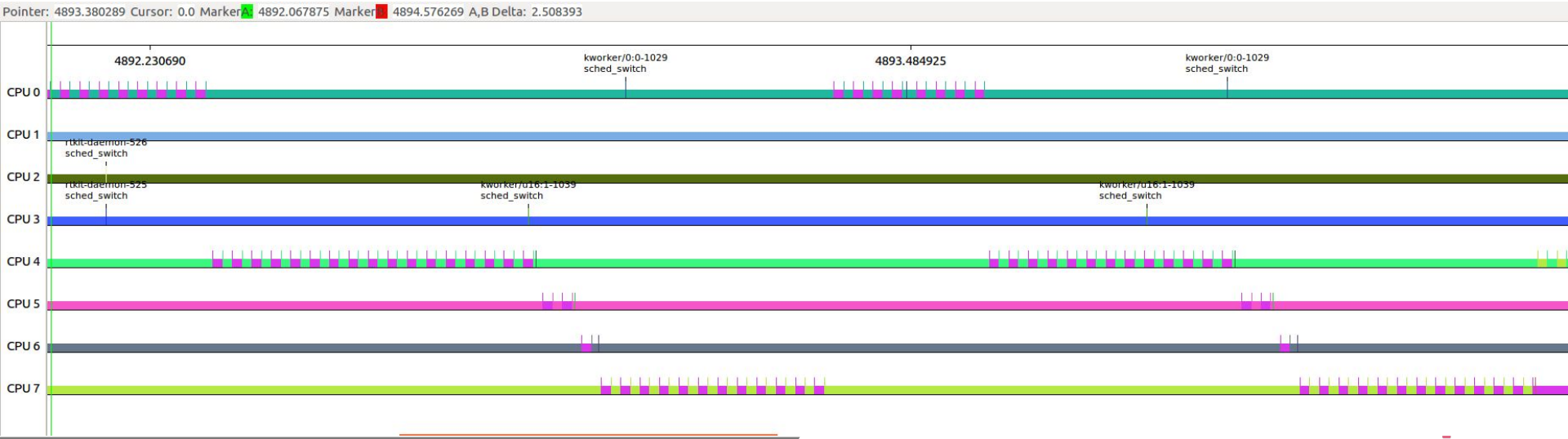
Open item #2 : detection of overloaded state

- Example of load balance stats for hackbench running on a dual quad core

	mainline					patchset				
			nr=0	nr=1	nr>1			nr=0	nr=1	nr>1
Total	1800		**312**	82		1144		**218**	56	
has_capacity	633	35%	312	82	239	314	27%	218	11	**85**
fully_busy						45	4%		45	**!!**
misfit	0	0%				0	0%			
asym_packing						0	0%			
overloaded	1167	65%			1167	785	69%			785

Open item #3 fairness

- How to ensure better fairness
  - N+1 tasks on N CPUs case
- What drives the migration ?
  - nr_balance_failed
  - load / 2 <= env->imbalance



Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contact@linaro.org