Rework load_balance

Vincent Guittot Linux Plumber Conference Scheduler MC 9 September 2019

01101110 01101101

101 01110101 01110010 01

CE CE CO 101

1110101 01110010

10 01111001

10011 00100000 01110100 c

1.01101000.0110

100101 01110010 011



Status

- Patchset on LKML
 - v2 <u>https://lkml.org/lkml/2019/8/1/529</u>
 - 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
 - \circ ~ 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



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
 - \circ N+1 tasks on N CPUs case
- What drives the migration ?
 - \circ nr_balance_failed
 - load / 2 <= env->imbalance

Pointer:	: 4893.380289 Cursor: 0.0 MarkerA: 4892.067875 Marker	6269 A,B Delta: 2.508393				
CPU 0	4892.230690	kworker/0:0-1029 sched_switch	4893.484925	kworker/0:0-1029 sched_switch		
CPU 1	rtkil-daemon-526 sched_switch					
CPU 2	rtxt-daemon-525 sched_switch	kworker/u10:1-1039 sched_switch		kworker/016:1-1039 sched_switch		
CPU 4						
CPU 5						
CPU 6						
CPU 7				000000		

Thank you

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

contact@linaro.org

10011 00100000 01110100 011011

11.01101000.01100

1100101 01110010 0110

01110101 01110010

10 01111001

101 01110101 01110010 01

M (1010)

11

