Rework load_balance

Vincent Guittot

20 May 2019

Agenda

- Introduction
- UCs
- Statistics
- Policy

Introduction

- Some UCs not correctly handle yet
- Others seem to be randomly supported

- Simplify calculate_imbalance
  - Remove old/meaningless heuristics
  - Remove weird imbalance calculation
    - Don’t set a value that “should” be enough to fix the imbalance

- Replace UC’s specific classification of group by generic tag

- Metrics have evolved
  - Runnable_load_avg
  - Util_avg
  - Load_avg

Some UCs

1 task per CPU

- 1 task per CPU with asymmetries
  - big Little

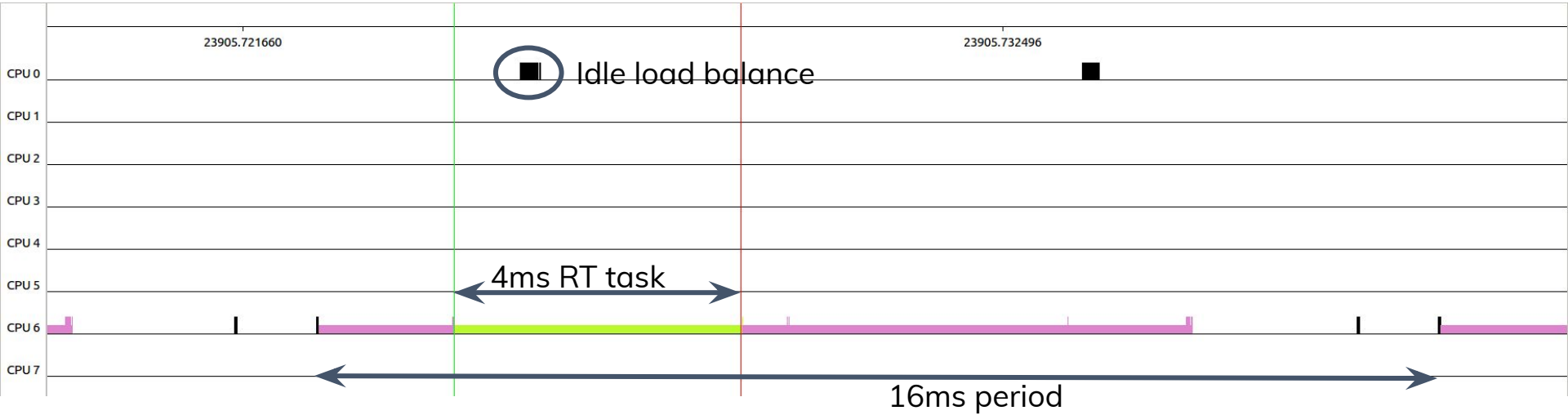
  - RT/DL/IRQ stolen time
  - Thermal pressure ?

fix_small_imbalance

- Depends of the weight of the tasks
- Depends of the initial state
  - Tasks placement
- Asymmetries screw up the default load balance mechanism
  - Compare load_avg

Class preemption

- CFS task preempted by RT but idle/newidle doesn't pull waiting task
  - Only takes into account cfs.h_nr_running



Running tasks

- Take into account tasks from other classes
  - Minimize resources contention

- Take into account sched_idle task
  - Improve scheduling latency

Ensure fairness

- Unbalanced system
  - 9 tasks on 8 CPUs

- When/How do we periodically migrate tasks to ensure fairness ?

Ensure fairness

- When does a task migrate ?
  - At every load_balance
  - with fix_small_imbalance

- Result looks a bit random
  - Use a more stable mechanism

Some UCs

- And probably lot of others

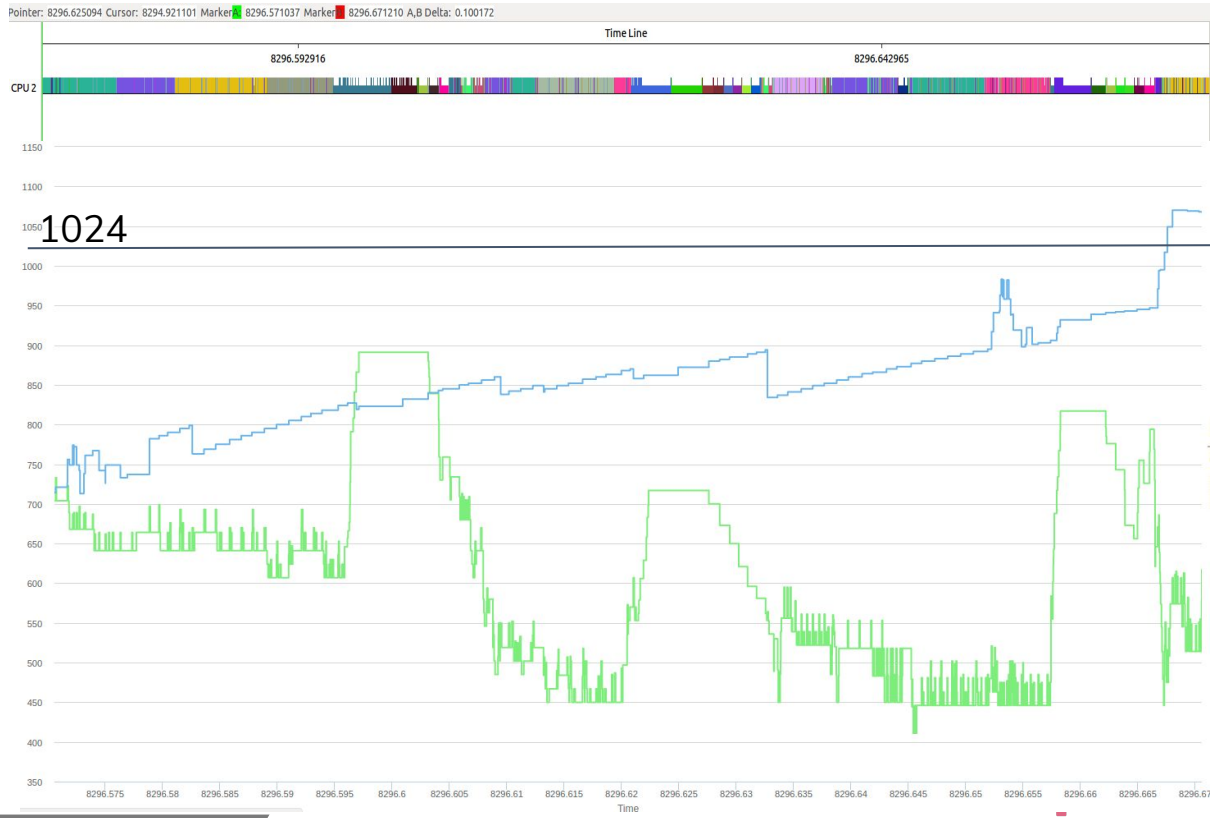
Statistics

Statistics

- running tasks
  - only nr_h_running is tracked
  - What about nr_running ?
  - What about CPUs with only sched_idle ?
- Utilization:
  - Util_avg is used to estimate spare capacity
  - But take care of jump due to migration
  - Util_est not always mitigate this effect
- Load :
  - Runnable is currently used
  - Lot of variance
  - Track both load and runnable_load ? similarly to find_idlest_group
- Capacity
  - Average available capacity: Max - IRQ/DL/RT and may be thermal

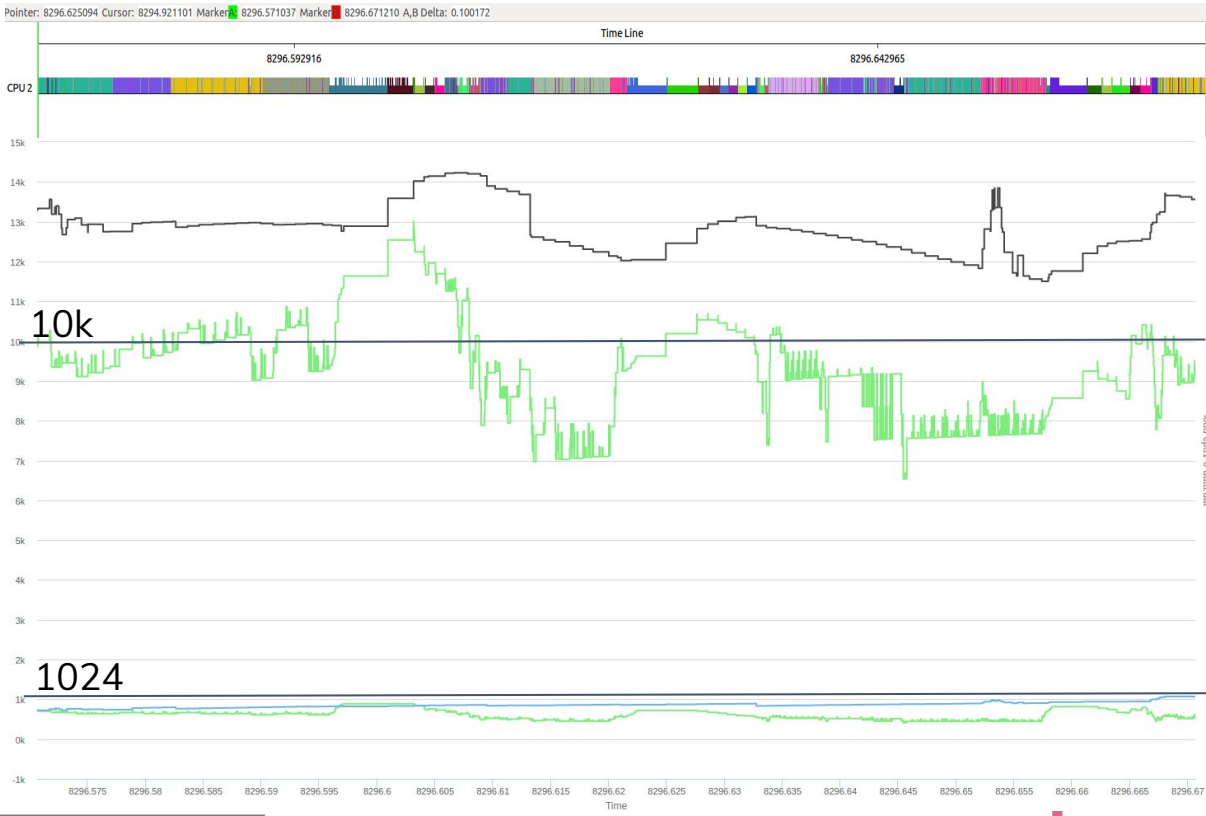
utilization

- Hackbench
  - CPUs are fully used
  - Lot of migration
- util_avg or util_est
  - Not always meaningful
- Take into runnable time ?
  - waiting time



runnable load

- runnable load quite high
- Lot of waiting time



Runnable vs load

- Runnable_load vs load
  - Runnable_load gives current running state
  - can use nr_running and utilization instead ?

- Remove weight from runnable
  - Track waiting time of tasks

Statistics

- Don’t bias statistics like with prefer sibling
  - Tag the group instead

- load_per_task
  - Doesn’t mean anything
  - Used when we don’t know how much to move
  - Should replaced by an explicit force migration

- sum_weighted_cpuload
  - Is the same as group_load since 4.19 and the disable of LB_BIAS
  - And remove cpu_load ...

Statistics

- nr_h_running
- nr_running
- sched_idle state
- Utilization : max(util_ag, util_est)
- Runnable_load_avg
- Load_avg
- capacity

Policy

Load balance sequence

- Load_balance
  - should_we_balance: select 1st idle or 1st CPU of the group
  - **find_busiest_group**(&env);
  - find_busiest_queue(&env, group);
  - Move waiting tasks or running task
  - Update balance interval

- Most of the work is done in
  - find_busiest_group
  - And find_busiest_queue to a lesser extend

Find busiest group

- find_busiest_group
  - **Update statistics and classify groups**
  - A list of conditions to decide if we are balanced
  - Calculate imbalance

Classify group

- Default state for the group
  - other
  - overloaded
- Plus some special cases
  - Imbalance
  - Misfit task
- Plus some short cut for asym packing
- Simplify the algorithm
  - Remove UC dedicated state

Group state

- Set a state for the group
  - Has capacity
  - Fully Busy
  - Overloaded

- Plus special cases
  - Fill an empty cpu
  - Move any task but at least one
  - Move a minimum load/util
  - Move a dedicated task

Policy

- Normal case: “overloaded”
  - we compute the load to move

- Other specific UCs where we want to move task
  - Fill an empty cpu
  - Unblocked a pinned related imbalance by moving any task
  - Move big task on big CPU
  - A dedicated task

- Imbalance can be a load or an utilization or a dedicated task

Thank you

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

contact@linaro.org