Improve utilization and load tracking in the scheduler

Presented by
Vincent Guittot

Date
November 3, 2016

Event
Linux Plumber Conference
Items

- Propagate tasks movement into task group
- Update idle CPUs
- Scale invariance
- Responsiveness of PELT
Propagate tasks movement into task group
Migrate inside task group

util_avg

util_avg CPU0

util_avg CPU1

migration is not seen at root level
Migrate in task group

Same behavior for load

load_avg

load_avg CPU0

load_avg CPU1

Time
Move between task groups

runnable_load_avg becomes null
Move between task groups

loaded CPU can be selected as idlest CPU even when other CPUs are idle

fork on CPU0

migrate on real idle CPU during next load balance
Propagate tasks movement into task group

- Propagate events down to the root domain
  - attach/detach task
  - remove task (asynchronous detach)
  - enqueue/dequeue task (already there)

- Propagate utilization
  - Simply add children changes to parent’s utilization

- Propagate load
  - Task group must not consumes more CPU than a task of equal weight
  - Scale task group’s load into the range of its shares
Propagate tasks movement into task group

Utilization migrates with task
Update idle CPUs
Blocked load/utilization of idle CPUs

load_avg/util_avg stay unchanged until next wake up or idle load balance

Impact the shares of busy CPUs
Update idle CPUs

- Ensure periodic update of blocked load
  - On an idle CPU like for periodic idle load balance
Scale invariance
Utilization and load of periodic task

- Utilization range is defined by
  - Number of running slots: \( r \)
  - And number of idle slots: \( i \)

\[
\frac{(1-y^r)}{(1-y^p)} \times y^i < \text{Utilization} < \frac{(1-y^r)}{(1-y^p)}
\]

- Similar for load
Scale utilization and load

- Invariance applies a scaling factor to the value of time slots
  - At max capacity the range is
    \[
    \frac{(1-y^r)}{(1-y^p)} \times y^i < \text{Utilization} < \frac{(1-y^r)}{(1-y^p)}
    \]
  - At lower capacity the range is
    \[
    C \times \frac{(1-y^{r'})}{(1-y^p)} \times y^{i'} < \text{Utilization} < C \times \frac{(1-y^{r'})}{(1-y^p)}
    \]
    with $C$ reflecting the compute capacity ratio
Scale utilization and load

Run 66ms at max capacity
Run 198ms at a 3rd of capacity

Scaled utilization for a task running 66ms every 330ms
Scale time

- Scale duration of time slot instead of scaling utilization of time slot
Scale time

Run always at a 3rd of capacity

Run 33ms at max capacity

Run 66ms at a half of capacity

Scaled utilization for a task running 33ms every 100ms
Impact on load/utilization

- Utilization/Load no more capped by current capacity
  - they can reach max value but need more time
- Utilization/Load of a task is fully invariant
  - Same min/max value at wake up/sleep whatever the compute capacity

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Max</th>
<th>Half (mainline)</th>
<th>Half (new)</th>
<th>A third (mainline)</th>
<th>A third (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Util(%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>972 (95%)</td>
<td>138ms</td>
<td>N/A</td>
<td>276ms</td>
<td>N/A</td>
<td>414ms</td>
</tr>
<tr>
<td>486 (47.5%)</td>
<td>30ms</td>
<td>138ms</td>
<td>60ms</td>
<td>N/A</td>
<td>90ms</td>
</tr>
<tr>
<td>256 (25%)</td>
<td>13ms</td>
<td>32ms</td>
<td>26ms</td>
<td>64ms</td>
<td>39ms</td>
</tr>
</tbody>
</table>
Responsiveness of PELT
Responsiveness of PELT

- Some figures
  - Time to reach 95%: 140ms
  - Time to reach 50%: 31ms

- Current decisions are proportional to utilization value
  - High Utilization value selects High OPP
  - Low Utilization value selects Low OPP

- Use other properties of utilization signal?
  - Use variation
    - An increase of 100 in one running time means that it has run for at least 5ms
  - Apply some kind of PID algorithm
    - Take into account variation of utilization
Other alternatives to improve responsiveness

- Use other scheduling class
  - deadline scheduler

- Run at fmax for some tasks
  - Boost utilization of some tasks and group of tasks

- Keep track of “peak per-entity utilization”
  - Track last max utilization value

- Use other load tracking mechanism