

Power Management Working Group – sched_mc Linaro connect Q4-11

Vincent Guittot <vincent.guittot@linaro.org>

https://wiki.linaro.org/WorkingGroups/PowerManagement/Specs/sche d_mc





Overview

- . CPU topology & sched_domain
- . Load balance & trigger
- Power saving load balance
- Quad cortex-A9
- . Dual cortex-A9
- Open issue
- Big.Little
- . New Load Balance



CPU topology & sched domain

- Dual/Quad cortexMP
 - New sched_domain configuration
 - . MC level instead of CPU level
 - Add SD_SHARE_PKG_RESOURCES flag





Load Balance

- Load balance monitoring
 - On running cores
 - On idle cores
- Load balance check on events
 - Newly idle cpu
 - Wake up task
 - Select an idle core with SD_SHARE_PKG_RESOURCES





Load Balance trigger

- Done at each sched_domain level
 - Only one for ARM cortex-A9 MP
- . Some requirements for load balancing:
 - Only on group which is out of capacity
 - Default core capacity is 1 thread
 - If there is an obvious imbalance (> imbalance_pct %)
- Asym packing
 - Load group with lowest cpu number 1st

Powersaving Load Balance

- Requirements :
 - At least 2 sched_groups
 - SD_POWERSAVINGS_BALANCE flag (CPU level only)
 - near idle group and a near full group
 - 0 < running threads < group capacity
 - group_capacity = group_power / SCHED_POWER_SCALE
- Not possible with Cortex-A9 MP system





- . Change cpu topology for power saving mode
 - Use arch_update_cpu_topology
 - Use CPU sched_domain level
 - with group_capacity > 1
- Emulate a virtual dual package:





- Pack tasks on one virtual package
 - . Low cpu load example :
 - Cyclictest with 10 threads
 - With default configuration
 - Tasks can be spread on 4 cores at wake up
 - With virtual dual packages
 - Tasks can be spread on 2 cores at wake up
 - Periodic load balance coould spread on 4 cores
 - Default config (without cpu topology patch)
 - Only periodic load balance will spread on 4 cores



sched_mc 0

Pointer: 571.294490 Cursor: 0.000000 MarkerA: 571.282010 Marker 571.293011 A,B Delta: 0.011000



sched_mc 2

Pointer:	1191	.06789	0 Cu	rsor: 0.00	00000 M	arker <mark>A</mark>	1191.	05837	6 Ma	rker <mark>B:</mark> 1	1191.	06938	2 A,BD	elta:	0.011006					
																	Time	Line		
											1191.066719									
CPU 0					I	I							I	I						
CPU 1			ľ																	-
CPU 2																				-
CPU 3																				
																				╵┖



- Use all cores
 - Heavy cpu load example :
 - Sysbench
 - . With default configuration
 - Tasks can be spread on 4 cores at wake up
 - With virtual dual package
 - Tasks can be spread on 2 cores at wake up
 - Periodic load balance will spread on 4 cores
 - Default config (without cpu topology patch)
 - Only periodic load balance will spread on 4 cores

sched mc 0

sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options: Number of threads: 12

Doing CPU performance benchmark

Threads started! Time limit exceeded, exiting... (last message repeated 11 times) Done.

Maximum prime number checked in CPU test: 10000

Test execution summary: 20.3001s total time: total number of events: 517 total time taken by event execution: 242.7256 per-request statistics: min: 315.64ms 469.49ms avg: 922.72ms max: approx. 95 percentile: 491.18ms

Threads fairness: events (avg/stddev): 43.0833/0.28 20.2271/0.05 execution time (avg/stddev):

sched mc 2

sysbench 0.4.12: multi-threaded system evaluation benchmark

Running the test with following options: Number of threads: 12

Doing CPU performance benchmark

Threads started! Time limit exceeded, exiting... (last message repeated 11 times) Done.

Maximum prime number checked in CPU test: 10000

Test execution summary: total time: 20.2956s total number of events: 528 total time taken by event execution: 242.4893 per-request statistics: 2.61ms

473.56ms

min:		372.61ms
avg:		459.26ms
max:		513.43ms
approx.	95 percentile:	473.5

Threads fairness:

44.0000/0.00 events (avg/stddev): 20.2074/0.04 execution time (avg/stddev):



Dual cortex-A9

- . Change cpu topology for power saving mode
 - Use CPU sched_domain level
 - with group_capacity > 1
- Increase cpu_power
 - . cpu_capacity = cpu_power / SCHED_POWER_SCALE
 - Use arch_scale_freq_power to increase cpu_power
 - Pull several tasks on 1 core
- . Emulate a dual package
 - Default config without cpu topolog







Dual cortex-A9

- Pack tasks on one core
 - . Low cpu load example :
 - Cyclictest with 10 threads
 - With default configuration
 - Tasks can be spread on 2 cores at wake up
 - With virtual dual package and increase of cpu_power
 - Periodic load balance could spread on 2 cores





Dual cortex-A9

sched_mc 0 • Pointer: 489.854568 Cursor: 0.000000 Marker 489.848411 Marker 489.858428 A,B Delta: 0.010016 Time Line 489.851270 489.856282 H H H 14 CPU 0 Þ UU. CPU 1

sched_mc 2

Pointer:	Pointer: 662.151906 Cursor: 0.000000 Marker 662.147882 Marker 662.157895 A,B Delta: 0.010012												
	Time Line												
			662.150742			662.155746							
CPU 0													
CPU 1	661.628713 <idle></idle>												
				6									





Dual Cortex-A9

- One cpu will be used while it has capacity
 - The trigger is the number of running threads
- And both cores when core0 is out of capacity
- . Heavy cpu load use case with few tasks
 - Use all cores
- Use cpufreq as a light cpu load detector
 - At lowest frequency, increase cpu_power and pull tasks
 - At other frequencies, use default cpu_power and spread tasks





Open issue

- Using cpu_power to pull tasks
 - Set a cpu_power to increase cpu_capacity is not advised
 - Intermediate step
- cpu_power update
 - Updated during "Idle" and "Not idle" load balance
 - But periodic load balance could not be called for a while
 - Cyclictest example
 - . RFC Patch available to ensure a periodic update inarc



Open issue

- . Idle load balance
 - The ILB call can be locked for a while
 - RFC patch available to solve such issue
 - Need to check new modifications around ILB call
- Spurious wake up
 - Idle Load balance called when nr_running > 1 on a cpu
 - No more true if cpu_power has been increased
 - To be studied and propose a patch





Big.Little

- cpu_power can be used to define asymetric system
 - More tasks will run on Big
- . How to differentiate heavy and light cpu load tasks ?
 - . Time weighted cpu load (see new load balance)
- How to differentiate background/foreground tasks ?
- How to differentiate IO task ?





New Load balance

- Load balance modification on going
 - Discussion during ELCE
- . Take into account different kind of topology
 - Dual/Quad cores (1 package)
 - Big.Little





Questions?

