openQA Tests - action #15668

[kernel][LTP][OpenQA] hyperthreading ht_interrupt won't run

2016-12-28 15:53 - rpalethorpe

Status:	Resolved	**Start date:**	2016-12-28
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Bugs in existing tests	**Estimated time:**	0.00 hour
Target version:	Done		
Difficulty:			

Description

Test fails with TCONF claiming system does not have HT enabled. However the CPU has the HT flag and shows 8 processors configured on a 4 core CPU.

The file /testcases/kernel/sched/hyperthreading/ht_interrupt/ht_utils.c (AFAICT) checks the /proc/cpuinfo file for the number of logical processors vs the number sockets/cores/threads and also looks for a line starting with cpu_package. If it finds cpu_package then it assumes that this is a Hyperthreading kernel. No such line is present on my system.

The other tests in the HT runfile, smt_smp_enable and smt_smp_affinity, run and pass, they have their own copies of ht_utils.c which are different.

Related issues:

Related to openQA Tests - action #15592: [LTP][OpenQA][kernel] LTP Native tes...	**Resolved**	**2016-12-20**

History

#1 - 2016-12-28 15:53 - rpalethorpe

- Related to action #15592: [LTP][OpenQA][kernel] LTP Native test runner minor issues added

#2 - 2017-01-04 10:39 - rpalethorpe

- Status changed from New to In Progress

Apparently this test requires 16 CPUs with two threads each. I can configure this in a VM, but ht_utils.c may still need updating for it to work. See commit: fbac41af9f2c5e34864b71f0eb94cdb3a2c08038

#3 - 2018-03-06 15:32 - sebchlad

- Subject changed from [LTP][OpenQA] hyperthreading ht_interrupt won't run to [kernel][LTP][OpenQA] hyperthreading ht_interrupt won't run

- Status changed from In Progress to Workable

#4 - 2018-03-26 15:07 - sebchlad

- Status changed from Workable to In Progress

- Assignee set to clanig

- Priority changed from Low to Normal

- Target version set to 448

As talked with Christian - current sprint.

#5 - 2018-03-27 14:31 - sebchlad

Christian, would be nice to have some update from you every 2-3 days, so other team members could easier help you if needed.

#6 - 2018-03-27 16:35 - clanig

Current Status:

What I found out
HT has been implemented in release 2.4.17. It has been handled as sort of SMP which means that it got displayed as there would have been many sockets instead of one socket with two logical threads per core.
With release 2.6.X (tbd) the whole thing changed. Now /proc/cpuinfo shows a physical id per socket, a core id for each core on the socket and a "processor" which means the logical processor id.

So when the program checks for HT before executing the test it always returns false because of this instability of /proc/cpuinfo.

The check works as follows:

1. Check whether the Kernel is capable to run HT.
2. Check whether the CPU is capable of HT.
3. Check whether HT is actually enabled.

This does make much sense so far. But I have some doubts:

1. I think it would be better to determine whether the Kernel is capable of HT by the version > 2.4.17 instead of a string in /proc/cpuinfo.
2. I can't complain about the way how the check for the capability of the CPU is made.
3. TBD: I think I should determine whether the current way to check for HT to be enabled covers all cases or whether there are cases when a logical vs physical core count of 2:1 is displayed although HT has been disabled by a kernel boot parameter for example. Probably there are cases that should be included.

What I have started to code so far:
Is retrieving the release version of the kernel where the test suite is executed in order to have a more reliable way to check for HT capability. And because my experience with C is more basic I had to look up some things to make progress with this first stage of the whole checking routine.

There is a corresponding issue on LTP git bugtracker:
https://github.com/linux-test-project/ltp/issues/211

#7 - 2018-04-03 07:20 - clanig

Update

All three checks don't work. I have fixed the check for an HT capable kernel determining it by the release version >2.4.17. Skipping these checks and just running the test seems to result in a fail with less cores but passed on a VM with many cores. I have looked into kernel sources to learn how they retrieve several information.

HT being just Intel's implementation of SMT it looks questionable to me whether it makes that much sense to have this test for HT which would restrict the number of threads per core to 2 while there are processors like the Power processors of IBM that can run in SMT mode with up to 8 threads.
Testing for a ratio of 2 threads per core is one of the key elements in the current checks whether HT is enabled.
I think I should find out how supported CPU architectures advertise their capabilities for SMT and if it's possible to combine the tests.

#8 - 2018-04-04 16:44 - clanig

It looks very much like the current kernel versions only provide file sockets for setting the processor affinity of the IRQ IDs in /proc/irq/ . These are used by the userspace daemon irqbalance.
According to this commit message the IRQ balancing ability of the kernel has been removed in October 2008:
https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git/commit/?id=8b8e8c1bf7275eca859fe551dfa484134eaf013b

The commit message is probably a bit irritating because it is only about 32 bit systems. But in the corresponding 64 bit files I haven't found the content that was removed in order to remove IRQ balancing at all.
I have also searched for other hints of a new implementation of IRQ balancing and so on but haven't found any.
HT capability was introduced with 2.4.17 and the IRQ balancing seems to have been removed with some release number like ~ 2.6.29. I haven't found out yet when IRQ balancing was introduced but probably not directly with 2.4.17. That would mean the HT_interrupt test would be obsolete for almost every Kernel.

As far as I understand correctly when interrupts are distributed decently on all available or at least several (logical) processors it can increase the overall performance and decrease the response latency for interrupts. But on the other hand real time applications will have increased latency when the interrupt is set to target their assigned processor.
So the ht_interrupt test is mainly focused on the performance of the kernel with ht in specific situations. That means there are no severe issues to be expected when the interrupts will target specific processors.

In my opinion, when all the information I gathered is correct the ht_interrupt test could be removed or tested and adapted to just run for these specific release versions when it's worth it.

#9 - 2018-04-05 07:23 - rpalethorpe

In my opinion, when all the information I gathered is correct the ht_interrupt test could be removed or tested and adapted to just run for these specific release versions when it's worth it.

I think removing it would be a good idea.

Maybe we should have a test which checks that setting the IRQ affinity works (if we do not have one already), but that would be a separate task.

#10 - 2018-04-05 09:46 - clanig

Thanks for your reply Richard,

we have smt_smp_affinity and smt_smp_enabled in the hyperthreading folder. Both seem to run fine on my machine. So I will prepare a PR that deletes the ht_interrupt files.

#11 - 2018-04-09 13:00 - clanig

- Status changed from In Progress to Resolved

The ht_interrupt test case has been removed.
https://github.com/linux-test-project/ltp/commit/28b8954b1a56418a1395932de57a09fc8ede6785

#12 - 2018-08-15 09:30 - sebchlad

- Assignee deleted (clanig)

- Target version changed from 448 to 445

Juest testing the stuff around and see how new project structure in progress works

#13 - 2018-08-17 07:54 - sebchlad

- Target version changed from 445 to Done