

## Performance of the High Level Trigger (Moore) on various CPU architectures

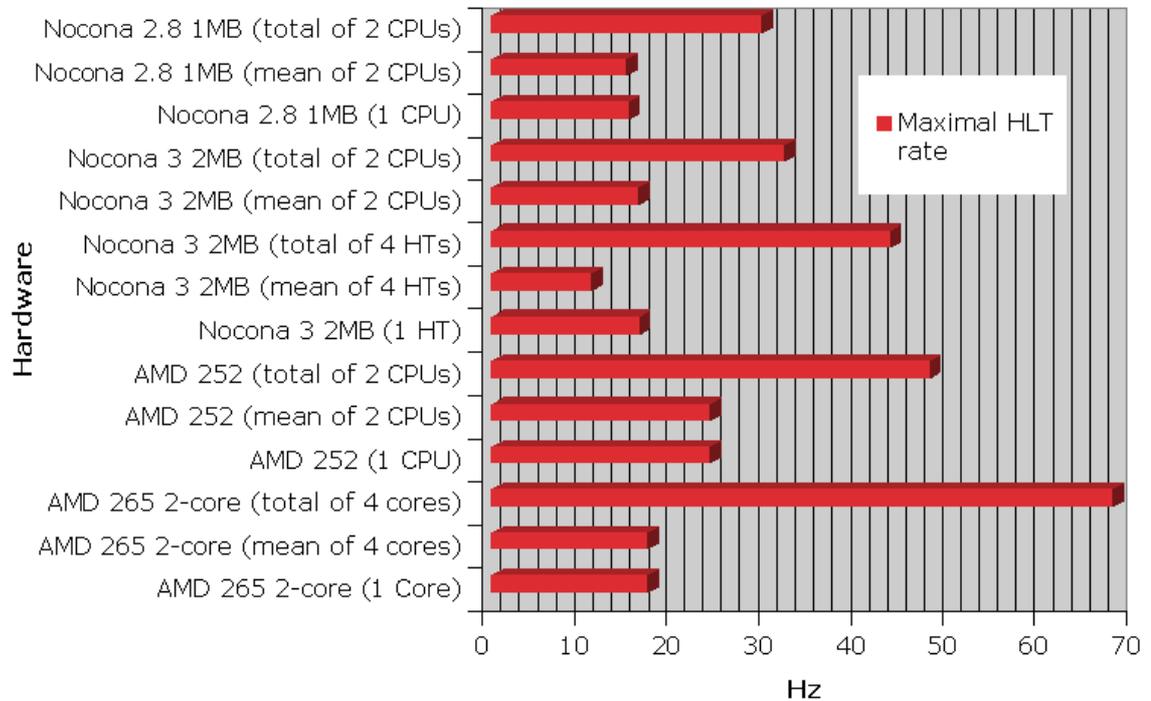
In order to see the influence of cache-sizes, multi-cores, hyper-threading and other new technologies, we have decided to run a realistic Moore job. This job consists of 50000 L1-accepted events, passed through Moore v1r2, Brunel is then run on the HLT accepted events. The sample has been created by running Euler v1r2 on L0 accepted RTTC events.

Definitions: In this page a CPU can have several cores (currently up to two) or several hyper-threads (currently up to two, only on the Intel architecture). All CPUs tested can run in 64-bit (x86\_64) mode, but have been booted with SLC3.0.4 32-bit. That is because the LHCb software is not yet ported to 64-bit.

The candidates tested were:

- A Dell 1425 SC with a dual Nocona Xeon processor with 1 MB cache at 2.8 GHz with 2 GB of DDR2 RAM, 800 MHz FSB "*Nocona 2.8*"
- A Dell 1425 SC with a dual Nocona Xeon processor with 2 MB cache at 3.0 GHz with 2 GB of DDR2 RAM, 800 MHz FSB "*Nocona 3*"
- A Dalco Dual AMD Opteron 252 processor with 1 MB cache at 2.6 GHz, 2 GB of DDR RAM (400 MHz) "*AMD 252*"
- A SuperMicro Dual AMD Dual-core Opteron 265 processor with 1 MB cache (per CPU) at 1.8 GHz, 4 GB of DDR RAM (400 MHz) "*AMD 265*"

**LHCb High Level Trigger  
("Moore on L1 accepted events / including Brunel") rate**



	AMD 265 2-core	AMD 265 2-core	AMD 265 2-core	AMD 252 (1 core)	AMD 252 (mean of 2)	AMD 252 (total of 2)	Nocona 3							
Maximal HLT rate	16.9	16.9	67.5	23.9	23.8	47.6	16	10.8	43.3	15.9	31.9	14.9	14.7	29.4

**Test details**

The LHCb High Level Trigger application MooreOffline was run using a "MarkusFile" (a flat file of raw-events in binary format) as an input to run in conditions as close as possible to true Online processing. This file contained 50000 L1 accepted events. To minimise overheads due to I/O the file was put into a RAM-disk. It turns out however that on all architectures the performance is totally dominated by the CPU. Running with the file on NFS gives practically identical results. All machines were booted with the same 32-bit SLC3.0.4 (i.e. modified RHEL 3) distribution. All machines were operated disk-less, i.e. the system resided on a NFS server. The same binary was used in all tests, it was once compiled on a server, which was itself not used in the tests. An "Moore" image needs roughly 157 MB when it is up and running, so there was plenty of RAM left on the machine, even considering the space needed for the RAM-disk.

**Analysis**

Clearly the AMD 265 Dual core has by far the best aggregated performance. Comparing single core and mean core performance on AMD processors show that for the LHCb trigger application, the scaling is very good (better than 5%).

## MoorePerformances < LHCb < TWiki

There is a clear advantage of the AMD Opteron architecture, even a single core running at 1.8 GHz is some 8% faster than a single Nocona at 2.8 GHz. Within an architecture the scaling seems to be roughly like the clock-speed. A single core on the AMD 252 is approximately 40% faster than a single core on the AMD 265. Similarly for the Xeon 2.8 and the Xeon 3.0, where it has to be borne in mind that the cache-size has doubled.

Hyperthreading is somewhat special. It is usually disabled on many machines because of the unpredictabilities it introduces particularly in the I/O handling. In this special test however it is clearly advantageous: it gives almost 45% performance gain on the non-HT mode (Hyperthreading was enabled and disabled via the BIOS). This not only saves the day somewhat for the Noconas, but also seems quite high - typical figures quoted by Intel are more in the 30% region. I think this is because, when running on multiple cores/hyperthreads we run the **same** application 4 times. This clearly helps with caching, in particular with the pages containing the text.

-- NikoNeufeld - 30 Aug 2005

---

This topic: LHCb > MoorePerformances

Topic revision: r3 - 2007-10-29 - StephenWotton



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback