

Process and Interrupt Affinity on Intel[®] Xeon[®] Processor E5 Servers with Intel[®] DDIO Technology

Application Note

April 2012

v1.0



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

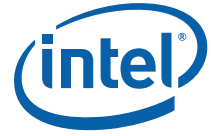
Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.*Other names and brands may be claimed as the property of others.

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

Copyright © 2012, Intel Corporation. All Rights Reserved.



Revision History

Date	Revision	Description
April 2012	v1.0	Initial release



The designs of the new Intel® Xeon® E5 processor servers have introduced some complexities that, if not accommodated in the application design, may not produce as large a performance gain as possible. Creating core affinity is critical in a Direct Data IO/NUMA environment. Whenever possible, processes that are identified as performance-critical should have affinity with the local socket connection, rather than remote socket connections. To that end, the most foolproof method for identifying the local and remote socket might be to refer to the block diagram of the server platform. This might not always be as easy as it is with the Intel Software Development Vehicle (SDV) code name: Rose City, where all the PCIE slots were branched off of a single socket/CPU1.

The example in the picture below shows the I/O riser module from a Dell R720* series server, an Intel Xeon E5 processor platform where the riser clearly identifies slot affinity to a given CPU or socket.



Figure 1. I/O Riser Identifies Slot Affinity to CPU or Socket

Another way to determine socket correlation would be to use APICID reference under `/proc/cpuinfo`, or possibly the CPU-Z detection engine for Windows*. (Assuming the APIC tables are implemented as defined in ACPI 5.0 specification and the operating system is able to read and parse the information for application consumption.) Once the local socket is identified and a determination is made about how to distribute the processes. (Keep in mind that simultaneous multi-threading (SMT) is especially effective when each thread is set to perform different types of operations and perform those operations with under-used CPU cycles.) Then, use the Thread Affinity Interface to assign the processes to the cores as needed.

One of the characteristics of the NUMA architecture is that each processor has its own localized memory module that it can access directly and thereby gain a distinct performance advantage. In addition, it can also access any memory module belonging to another processor using the QPI link. However, that memory access across the QPI link carries a performance/latency penalty in relation to a memory access on a local memory bus. Therefore, optimal performance would require adherence to the strategies for NUMA optimization outlined in the [Optimizing Software Applications for NUMA](#) whitepaper.

For specifics about how to pin the processes to the chosen cores, see the [82575,82576, 82598, 82599 Ethernet Controllers Interrupts Application Note](#).

§ §