



# SQL Server™ on NEC Express5800/A1080a Server Best Known Methods

NEC Corporation of America  
Information Technology Group

June 2010

Recommendations presented in this document should be thoroughly tested before being implemented in a production environment. While recommendations are listed in an itemized format, they should not be considered standalone. All recommendations have dependencies and implementation of any recommendation in isolation can be detrimental to performance.

Information in this document is subject to change without notice. Complying, with all applicable copyright laws is the responsibility of the user. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of NEC Corporation of America.

© Copyright 2010 by NEC Corporation of America. All rights reserved.

NEC is a registered trademark and Empowered by Innovation is a trademark of NEC Corporation.

Windows, Windows NT, Windows 2008, SQL Server 2008 are either registered trademarks or trademarks of Microsoft Corporation in the USA and other countries.

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

All other trademarks are the property of their respective owners.

# Table of Contents

- Overview ..... 4**
- Express5800/A1080a Server Overview ..... 4**
- Express5800/A1080a Configuration..... 4**
- Hardware Configuration..... 4**
- BIOS Performance Parameters ..... 4**
- Windows Operating System and SQL Server Configuration..... 5**
- Windows Operating System Services Trimming ..... 5**
- Intel Ethernet RSS queues, Interrupt Moderation and RX/TX buffer settings..... 5**
- Enable Large Page Support..... 5**
- SQL Soft-NUMA ..... 6**
- SQL Max Server Memory..... 6**
- SQL Max Worker Threads ..... 6**
- SQL Lightweight Pooling Option ..... 7**
- SQL Max Degree of Parallelism ..... 7**
- SQL Recovery Interval ..... 7**
- Command Line options for SQL Server and SQL Trace Flags..... 8**

# Overview

## Express5800/A1080a Server Overview

NEC's Express5800/A1080a series server can accommodate up to 2 TB of memory and 128 threads with modular in-box partitioning in an innovative single 7U chassis. Whether for memory hungry virtualization solutions, compute intensive databases, or high availability for mission-critical business, IT managers have the flexibility to scale and configure the new Express5800/A1080a series server for their unique datacenter requirements.

## Express5800/A1080a Configuration

### Hardware Configuration

| Component | Should Be:  |
|-----------|---|
| Processor | For each model, choose SKU with largest cache size, highest QPI link speeds, and highest CPU frequency. Highest performing model and SKU:<br>Intel® Xeon® Processor X7560 (24M Cache, 2.26 GHz, 6.40 GT/s Intel® QPI)   |
| Memory    | Spread 1067 MT/s DDR3 DIMMs equally across all DDR3 channels.<br><br>It is important to note that the number of DIMMS populated per channel (DPC) will have an impact on performance. Latency to memory increases as higher DIMMs per channel are used. As a general rule, it is best to populate all memory sockets equally with the highest supported frequency DIMMs.<br><br>Enable NUMA.<br><br><i>Non-uniform Memory Architecture (NUMA)</i> is a system memory design such that physical processors are tightly associated with their own, local physical memory. This can provide reduced memory latencies as frequently accessed structures are fetched from a memory controller's local memory. Many operating systems can make use of NUMA by allocating memory structures in "local" memory. In addition many enterprise applications such as database management systems can organize memory to take advantage of NUMA. |
| I/O       | Use PCI-e 2.0 based devices. PCI-Express slot configurations vary enough per system to make a general recommendation. Be sure to populate the PCI-E x16 slots (slot 7 & 13) first followed by PCI-E x8 slots.   |

### BIOS Performance Parameters

| Feature                        | Values (Default in bold) | When to try it   |
|--------------------------------|--------------------------|--|
| Hyper-Threading Technology     | <b>Enable</b>            | For any workload that can make use of more cores                                     |
| Turbo Boost Technology         | <b>Enable</b>            | For any processors and workloads that can take advantage of higher core frequencies. |
| Hardware Prefetcher            | <b>Enable</b>            | For any processors: Workloads with sequential or predictable data access;            |
| Adjacent Cache Line Prefetcher | <b>Enable</b>            | For any processors: Workloads with sequential or predictable data access;            |

# Windows Operating System and SQL Server Configuration

## Windows Operating System Services Trimming

As a general rule it is best to run the operating system with as few services as possible. The first places to look are for any non-Microsoft services and disable those. Of course be very careful about what services you disable as some are required for normal operation of the system.

## Intel Ethernet RSS queues, Interrupt Moderation and RX/TX buffer settings

References:

<http://www.intel.com/network/connectivity/products/whitepapers/318483.pdf>

<http://www.intel.com/support/network/adapter/pro100/sb/CS-029402.htm>

Some of these settings require some experimentation; however here are some rules of thumb:

- Increasing the RSS queues to the maximum value will should spread networking related interrupts across as many CPUs as possible.
- Interrupt moderation set to 'extreme' should help reduce the number of interrupts related to networking traffic.
- Increasing the number of RX and TX buffers can prevent dropped packets related to not enough buffers available for either send or receive.

## Enable Large Page Support

### Lock Pages in Memory

Reference: <http://msdn.microsoft.com/en-us/library/ms190730.aspx>

To enable the lock pages in memory option

1. On the Start menu, click Run. In the Open box, type gpedit.msc.  
The Group Policy dialog box opens.
2. On the Group Policy console, expand Computer Configuration, and then expand Windows Settings.
3. Expand Security Settings, and then expand Local Policies.
4. Select the User Rights Assignment folder.  
The policies will be displayed in the details pane.
5. In the pane, double-click Lock pages in memory.
6. In the Local Security Policy Setting dialog box, click Add.
7. In the Select Users or Groups dialog box, add an account with privileges to run sqlservr.exe.

## SQL Trace Flag 834

Reference: <http://support.microsoft.com/kb/920093>

### Trace flag 834: Use Microsoft Windows large-page allocations for the buffer pool

Trace flag 834 causes SQL Server to use Microsoft Windows large-page allocations for the memory that is allocated for the buffer pool. The page size varies depending on the hardware platform, but the page size may be from 2 MB to 16 MB. Large pages are allocated at startup and are kept throughout the lifetime of the process. Trace flag 834 improves performance by increasing the efficiency of the translation look-aside buffer (TLB) in the CPU.

## SQL Soft-NUMA

### CPU affinity

Reference: <http://msdn.microsoft.com/en-us/library/ms345357.aspx>

To configure SQL Server to use software non-uniform memory access (soft-NUMA), you must edit the registry to add a node configuration affinity mask. The soft-NUMA mask can be stated as a binary or DWORD (hexadecimal or decimal) registry entry. You must restart the Database Engine to configure soft-NUMA. Before configuring software NUMA, see [Understanding Non-uniform Memory Access](#) and [How SQL Server Supports NUMA](#). See reference for more details.

### IO affinity

Reference: <http://msdn.microsoft.com/en-us/library/ms345346.aspx>

TCP/IP port to non-uniform memory access (NUMA) node affinity is configured as a server setting in SQL Server Configuration Manager. See reference for more details.

## SQL Max Server Memory

Reference: <http://msdn.microsoft.com/en-us/library/ms178067.aspx>

Use max server memory to prevent the SQL Server buffer pool from using more than the specified amount of memory, thus leaving remaining memory available to start other applications quickly. SQL Server does not immediately allocate the memory specified in max server memory on startup. Memory usage is increased as needed by SQL Server until reaching the value specified in max server memory. SQL Server cannot exceed this memory usage unless the value of max server memory is raised.

## SQL Max Worker Threads

Reference: <http://msdn.microsoft.com/en-us/library/ms187024.aspx>

Use the max worker threads option to configure the number of worker threads available to Microsoft SQL Server processes. SQL Server uses the native thread services of the Microsoft Windows 2000 and Windows Server 2003 operating systems so that one or more threads support

each network that SQL Server supports simultaneously, another thread handles database checkpoints, and a pool of threads handles all users.

Please refer to reference on specific guidelines for this setting.

## SQL Lightweight Pooling Option

Reference: <http://msdn.microsoft.com/en-us/library/ms178074.aspx>

Use the **lightweight pooling** option to provide a means of reducing the system overhead associated with the excessive context switching sometimes seen in symmetric multiprocessing (SMP) environments. When excessive context switching is present, lightweight pooling can provide better throughput by performing the context switching inline, thus helping to reduce user/kernel ring transitions.

Recommend to set this sp\_configure value to 1.

## SQL Max Degree of Parallelism

Reference: <http://msdn.microsoft.com/en-us/library/ms181007.aspx>

When SQL Server runs on a computer with more than one microprocessor or CPU, it detects the best degree of parallelism, that is, the number of processors employed to run a single statement, for each parallel plan execution. You can use the **max degree of parallelism** option to limit the number of processors to use in parallel plan execution. The default value of 0 uses all available processors.

Recommend to set this sp\_configure value to 1.

## SQL Recovery Interval

Reference: <http://msdn.microsoft.com/en-us/library/ms190770.aspx>

Use the **recovery interval** option to set the maximum number of minutes per database that Microsoft SQL Server needs to recover databases. Each time an instance of SQL Server starts, it recovers each database, rolling back transactions that did not commit and rolling forward transactions that did commit but whose changes were not yet written to disk when an instance of SQL Server stopped. This configuration option sets an upper limit on the time it should take to recover each database. The default is 0, indicating automatic configuration by SQL Server. In practice, this means a recovery time of less than one minute and a checkpoint approximately every one minute for active databases.

Recommend to set this sp\_configure value to 32767.

## Command Line options for SQL Server and SQL Trace Flags

### Trace Flags used for OLTP benchmarks

Reference: <http://support.microsoft.com/kb/920093>

Suggest to use the following command line for SQL server for the TPC-E (OLTP) benchmark: -x -c -T3502 -T834 -T8744 -T661 -T652

#### **-c**

Indicates that an instance of SQL Server is started independently of the Windows NT Service Control Manager. This option is used when starting SQL Server from a command prompt, to shorten the amount of time it takes for SQL Server to start. (Note that when you use this option, you cannot stop SQL Server by using SQL Server Service Manager or the net stop command, and if you log off the Microsoft Windows NT® system, SQL Server will be stopped.)

#### **-x**

Disables maintaining CPU statistics

#### **-T 3502**

You can set trace flag 3502 to include a message in the log at the beginning and ending of each checkpoint. This trace flag may be useful if you use trace flag 3505.

#### **-T 834: Use Microsoft Windows large-page allocations for the buffer pool**

Trace flag 834 causes SQL Server to use Microsoft Windows large-page allocations for the memory that is allocated for the buffer pool. The page size varies depending on the hardware platform, but the page size may be from 2 MB to 16 MB. Large pages are allocated at startup and are kept throughout the lifetime of the process. Trace flag 834 improves performance by increasing the efficiency of the translation look-aside buffer (TLB) in the CPU.

Trace flag 834 applies only to 64-bit versions of SQL Server. You must have the Lock pages in memory user right to turn on trace flag 834. You can turn on trace flag 834 only at startup.

Trace flag 834 may prevent the server from starting if memory is fragmented and if large pages cannot be allocated. Therefore, trace flag 834 is best suited for servers that are dedicated to SQL Server.

#### **-T 8744: Disable pre-fetching for ranges**

Trace flag 8744 disables pre-fetching for the Nested Loops operator. Incorrect use of this trace flag may cause additional physical reads when SQL Server executes plans that contain the Nested Loops operator. For more information about the Nested Loops operator, see the "Logical and physical operators reference" topic in SQL Server 2005 Books Online.

You can turn on trace flag 8744 at startup or in a user session. When you turn on trace flag 8744 at



startup, the trace flag has global scope. When you turn on trace flag 8744 in a user session, the trace flag has session scope.

#### **-T 652: Disable page pre-fetching scans**

Trace flag 652 disables page pre-fetching during scans. You can turn on trace flag 652 at startup or in a user session. When you turn on trace flag 652 at startup, the trace flag has global scope. When you turn on trace flag 652 in a user session, the trace flag has session scope. If you turn on trace flag 652, SQL Server no longer brings database pages into the buffer pool before these database pages are consumed by the scans. If you turn on trace flag 652, queries that benefit from the page pre-fetching feature exhibit low performance.

#### **-T 661: Disable the ghost record removal process**

Trace flag 661 disables the ghost record removal process. A ghost record is the result of a delete operation. When you delete a record, the deleted record is kept as a ghost record. Later, the deleted record is purged by the ghost record removal process. When you disable this process, the deleted record is not purged. Therefore, the space that the deleted record consumes is not freed. This behavior affects space consumption and the performance of scan operations.

When you turn on trace flag 661 at startup or in a user session, trace flag 661 always applies across the server and has global scope. If you turn off this trace flag, the ghost record removal process works correctly.

### **OLAP specific Trace Flags**

Reference: <http://support.microsoft.com/kb/920093>

#### **-T 2301: Enable advanced decision support optimizations**

Trace flag 2301 enables advanced optimizations that are specific to decision support queries. This option applies to decision support processing of large data sets.

You can turn on trace flag 2301 at startup or in a user session. When you turn on trace flag 2301 at startup, the trace flag has global scope. When you turn on trace flag 2301 in a user session, the trace flag has session scope.