Оптимизация многопоточной обработки сетевого трафика

17 Фев 2016

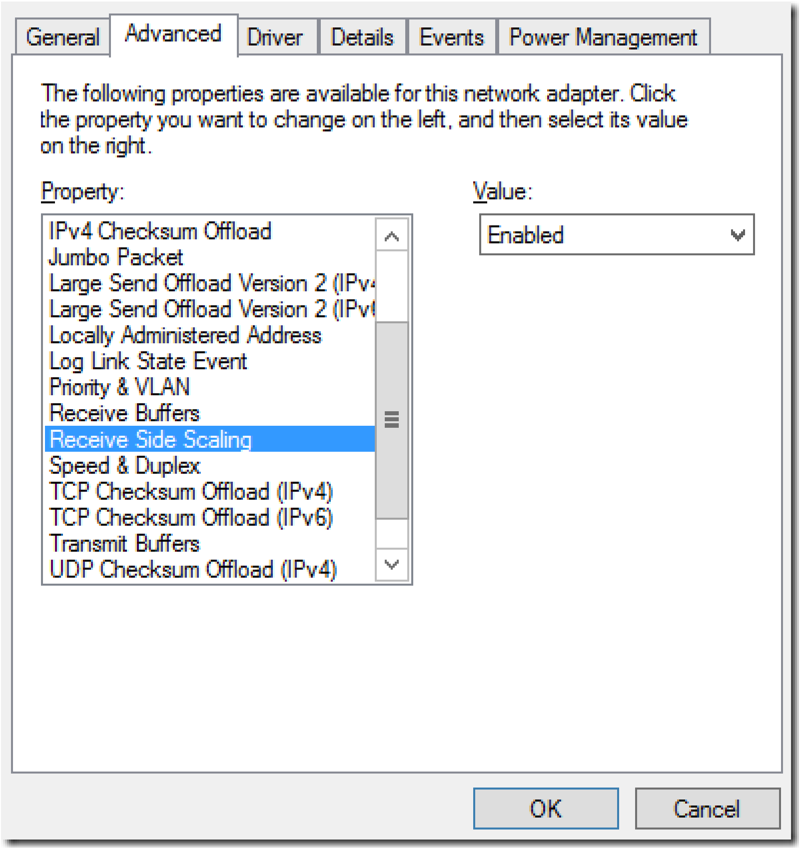
Михаил Закусило

Технология RDMA аппаратно подкреплена способностью адаптера работать в многопоточном режиме. Это означает, что определенное количество программных тредов (**threads** – *потоков* или, по другой терминологии, *очередей* – **queues**), генерируемых коммуникационным контроллером, подключенным к PCIe-шине, должны быть обработаны соответствующим количеством процессорных ядер. Давайте детальнее остановимся на том механизме, который устанавливает соответствие между потоками сетевого трафика и ядрами процессора. Но прежде отметим, что на платформах с двумя или более процессорными гнездами, а также на платформах поддерживающих технологию Cluster-on-Die (а это CPU, как минимум, с 10 ядрами) необходимым условием обслуживания многопоточности для bus master устройств является наличие специальной поддержки со стороны ACPI.

Напомним, что Receive Side Scaling – это механизм Microsoft Server 2012/R2, позволяющий эффективно разделить обработку входного сетевого трафика между несколькими ядрами мультипроцессорной системы и установить соответствие между соединениями и процессорами, обслуживающими данные соединения. При этом фигурируют следующие (иногда взаимно противоречивые) критерии:

1. Обслуживание трафика RNIC должны выполнять процессоры NUMA-узла, ближайшего к PCIe-подключению данного NIC;
2. Нагрузку в системе желательно равномерно распределить между всеми процессорами всех NUMA-узлов;
3. Обслуживание одного соединения (*TCP connection*) желательно возложить на один процессор (или минимальное количество процессоров), так как в кэш-памяти этого процессора будет находиться контекст, связанный с обслуживанием данного потока. Переключение на другой процессор приведет к необходимости повторного кэширования контекста данного потока и сделает бесполезными ранее кэшированные данные.

По умолчанию, технология Receive Side Scaling не обязательно будет активизирована в свойствах сетевого контроллера. Убедиться в этом можно заглянув в закладку «Advanced»:

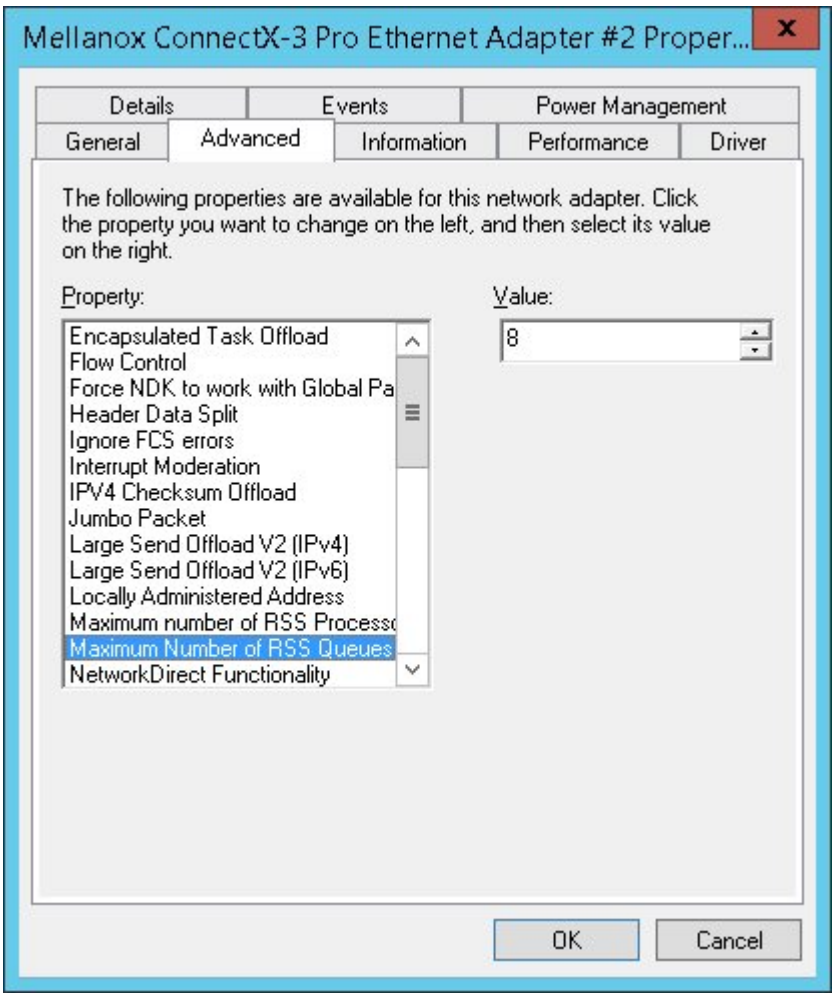


Альтернативный путь – использование команд PowerShell **Enable-NetAdapterRss -Name** "*имя_адаптера*" с последующей проверкой статуса **Get-NetAdapterrss -Name** "*имя_адаптера*".

```
PS C:\Windows\system32> Get-NetAdapterRss -name "Ethernet 6"


Name                                            : Ethernet 6
InterfaceDescription                            : Mellanox ConnectX-3 Pro Ethernet Adapter #2
Enabled                                         : True
NumberOfReceiveQueues                           : 8
Profile                                         : Closest
BaseProcessor: [Group:Number]                   : 0:20
MaxProcessor: [Group:Number]                    : 0:38
MaxProcessors                                   : 8
RssProcessorArray: [Group:Number/NUMA Distance] : 0:20/0  0:22/0  0:24/0  0:26/0  0:28/0  0:30/0  0:32/0  0:34/0
                                                  0:36/0  0:38/0
IndirectionTable: [Group:Number]                : 0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
                                                  0:20    0:22    0:24    0:26    0:28    0:30    0:32    0:34
```

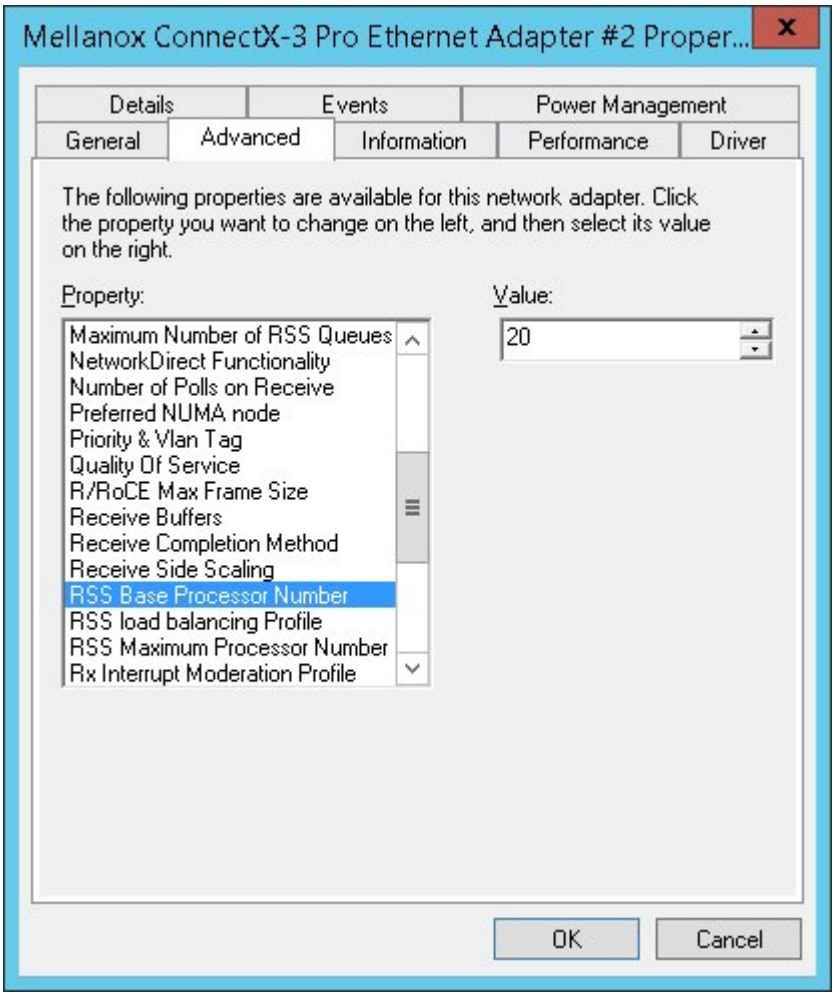
Давайте разбираться параметрами: как ими управлять и что какой означает? Начнем с ***NumberOfReceiveQueues*** (в свойствах адаптера ему соответствует пункт меню *Maximum number of RSS queues*):



Этот параметр определяет, какое количество потоков мы можем поручить RNIC-адаптеру. Здесь важным является только предельное значение: в нашем примере Mellanox ConnectX-3 аппаратно ограничен восемью соединениями. При необходимости и в зависимости от имеющихся процессорных ресурсов эту цифру можно уменьшить до 1-2 потоков. Не менее важно знать, что максимальное значение **доступно** пользователю. Практика показывает, что неактуальные драйверы порой некорректно формируют *Maximum number of RSS queues*. Были случаи, когда этот параметр для ConnectX-3 искусственно ограничивался 4 очередями.

Параметр *RSS Base processor number*

В консольном варианте этот параметр представлен строкой ***BaseProcessor: [Group:Number]*** и означает минимальный (стартовый) номер обслуживающего процессора, назначенный RNIC-адаптеру при инициализации (как правило, выбирается по критерию NUMA-оптимальности).

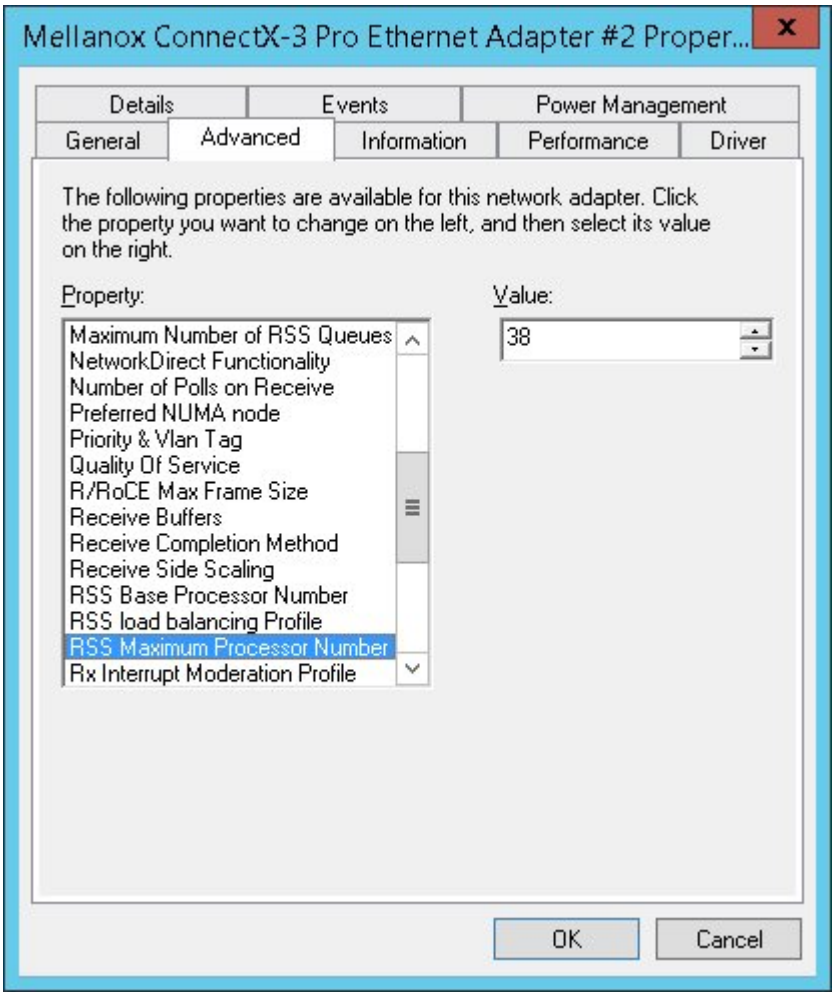


Здесь ***Group*** – понятие, введенное компанией Microsoft для систем, содержащих более 64 логических процессоров. Это связано с тем, что модель Win64 API базируется на 64-битных параметрах. Для перечисления большего, чем 64 числа ядер, когда каждому процессору соответствует один бит, требуется деление на группы, по 64 процессора в каждой. Последняя из групп может быть до конца не заполненной.

Исходя из сказанного, на скромно оснащенных платформах все процессоры уместятся в нулевую группу, а нумерация ядер начнется с BSP-процессора и продолжится на AP1, AP2 etc.

Параметр *RSS Maximum processor number*

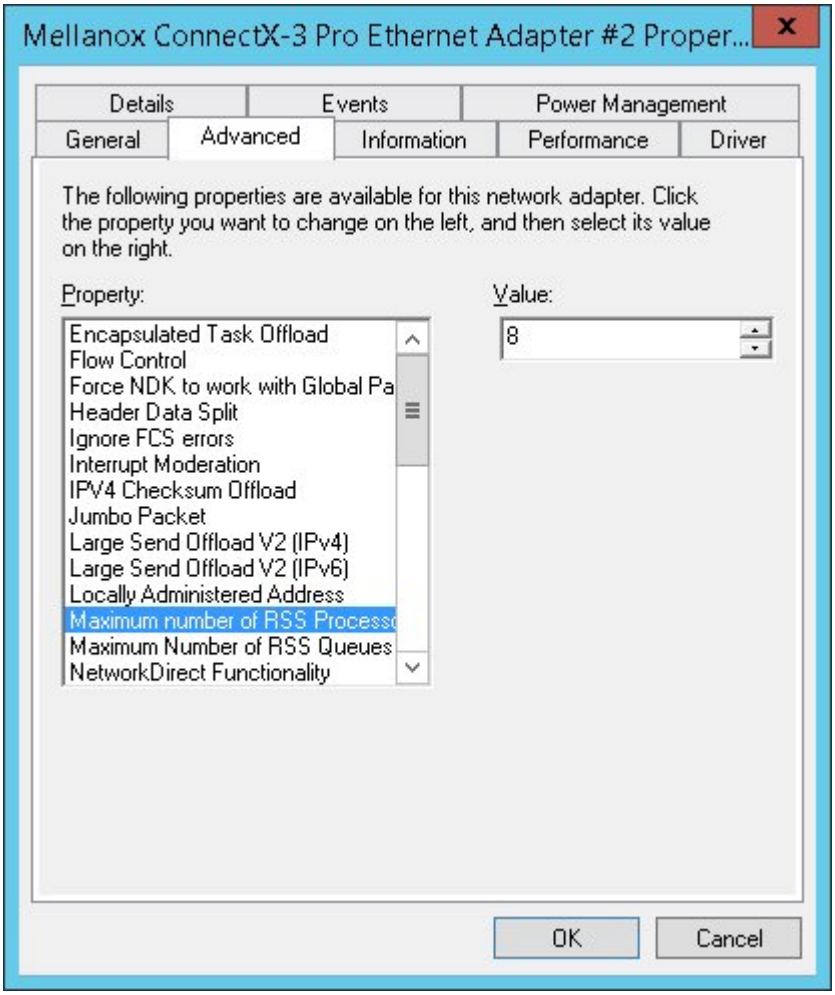
В консольном варианте ***MaxProcessor: [Group:Number]*** – максимальный (последний) номер процессора, назначенный RNIC-адаптеру при инициализации. Все аналогично предыдущему параметру, но с одним существенным уточнением, важным для дальнейшего понимания.



На MPS-платформе, где используется Receive Side Scaling, в расчет принимаются только физические процессорные ядра! Технология Hyper-Threading не влияет на многопоточное обслуживание TCP-соединений. Поэтому при включенной HT-технологии нумероваться будут только четные ядра. По хорошему, ее стоило бы просто [отключить в UEFI Setup](#).

Параметр *Maximum number of RSS processors*

Данный параметр, представленный в консоли PowerShell как ***MaxProcessors***, определяет количество процессоров, которое операционная система может выделить для обслуживания данного адаптера NIC. Номера этих процессоров должны входить в диапазон, ограниченный параметрами ***BaseProcessor*** и ***MaxProcessor***.

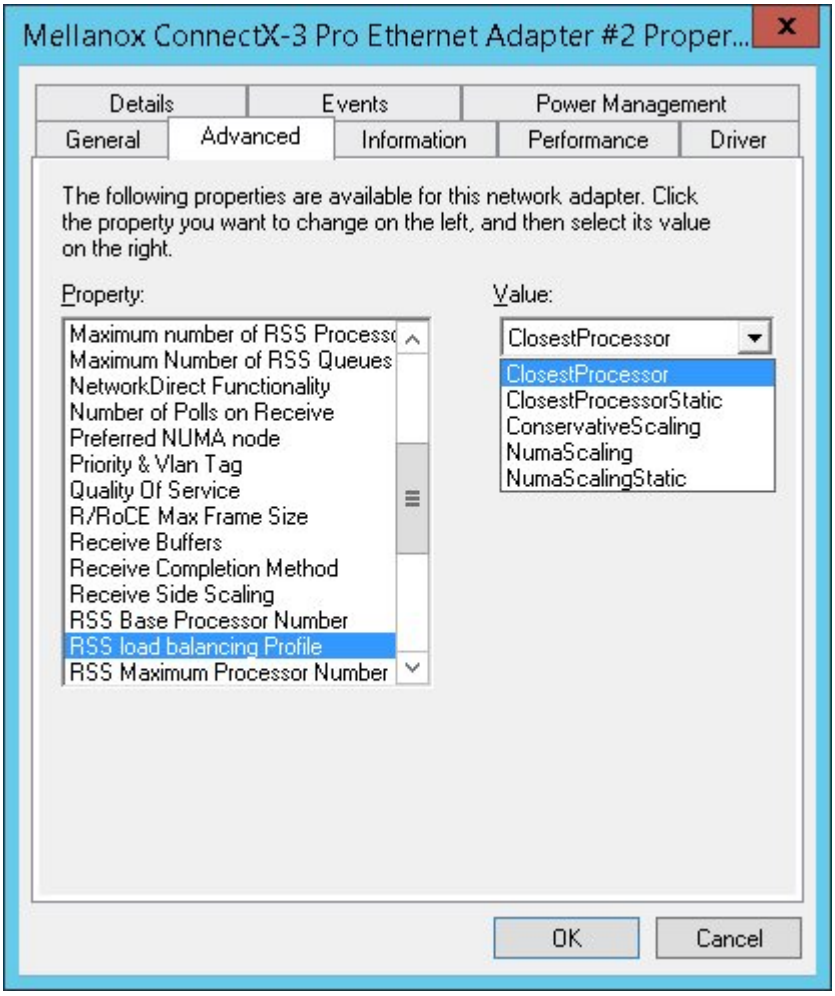


Реагируя на изменения сетевой нагрузки, программное обеспечение, оптимизирует использование процессоров bus-master адаптером, действуя в рамках ограничений, задаваемых указанными параметрами. При этом программное обеспечение может:

1. Переключать нагрузку и добавлять процессоры из того же NUMA узла, если установленный профиль задает оптимизацию по критерию минимальной NUMA-дистанции (профили *Closest, ClosestStatic*);
2. Переключать нагрузку и добавлять процессоры из других NUMA узлов, если данный профиль задает оптимизацию по критерию балансировки нагрузки между NUMA-узлами (профили *NUMA, NUMAStatic*);
3. Использовать минимальное количество процессоров, если данный профиль задает минимизацию количества процессоров (профиль *Conservative*).

Параметр *RSS Load balancing profile*

Параметр *RSS Load balancing profile* в консоли сокращен до прозаичного и понятного ***Profile*** и позволяет задействовать один из перечисленных выше сценариев.



ClosestProcessor

Выбирает для обслуживания заданного RNIC-адаптера, процессоры, ближайшие к этому адаптеру с точки зрения NUMA-топологии.

ClosestProcessorStatic

Нагрузка разделена между процессорами по фиксированной схеме, принятой при старте. Критерии, как и в предыдущем сценарии, но динамическое изменение балансировки не используется.

NUMAScaling

Распределяет потоки трафика по критерию оптимальной балансировки нагрузки по NUMA-узлам. Возможно, при этом приносится в жертву требование минимизации топологической дистанции между PCIe-линком, используемым RNIC и целевым блоком памяти в пользу более равномерного распределения нагрузки между узлами. Интересно, что в презентациях от Microsoft этот сценарий иногда называется ***NUMA Dynamic***.

NUMAScalingStatic

Как и предыдущий сценарий использует NUMA-технологию, но без динамического изменения балансировки нагрузки. Требование минимизации топологической дистанции между PCIe-линком и целевым блоком памяти выполняется в той степени, как это исходно определено таблицей соответствия между потоками трафика и логическими процессорами, используемыми для их обработки. В ряде источников профиль ***NUMA Static***принимается за сценарий по умолчанию.

ConservativeScaling

Минимизация количества используемых процессоров. Поскольку количество обработанных запросов останется тем же, утверждение об уменьшении количества прерываний можно трактовать по-разному, например:

- уменьшение количества сообщений, передаваемых между процессорами;
- упрощение формата сообщений **MSI** (*Message Signaled Interrupts*), что связано с уменьшением количества адресуемых процессоров, а также оптимизации работы кэш-памяти процессора, обслуживающего трафик.

Теги: Intel, RDMA