



# FCoE Host Side Considerations

Diego Crupnicoff – Mellanox Technologies  
Matthew Gaffney/Ariel Hendel – Sun Microsystems  
August 2007

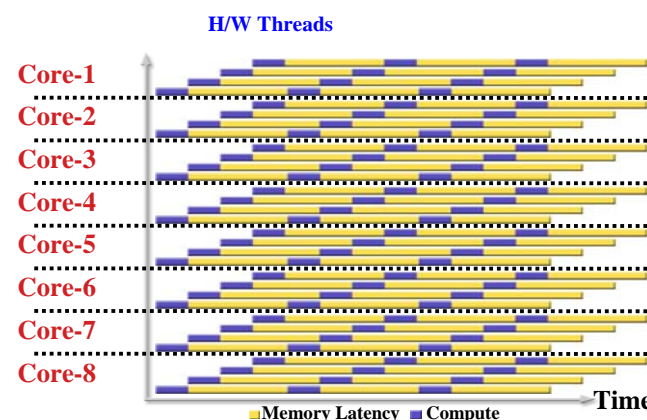
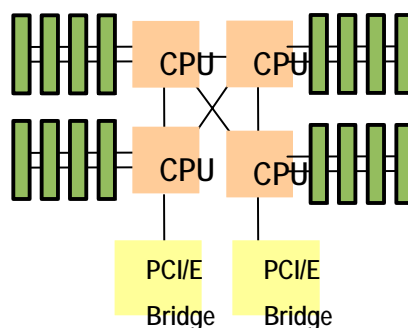
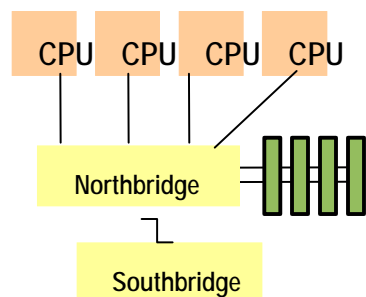


# Scope

- This presentation covers first order effects of FCoE header choices on host interface devices and software stacks
- The perspective is based on generic common practice and trends around:
  - > Data Path Queuing
  - > Resource isolation
  - > Virtualization and partitioning
  - > Software stacks

# Resource Isolation

- Resource isolation has been introduced to network interfaces for the purpose of:
  - > Matching the computational parallelism of the host
  - > Minimize blocking
  - > Create the scaffolding for I/O virtualization (at high speed)



# Virtualization and Partitioning

- Multiple OS instances per system
  - > Number of instances expected to be proportional to threads (not just cores)
  - > 64 threads per socket in 2007
- Virtual network interface addressing and device sharing mechanisms have some limitations
  - > Because they lie at the administrative boundary between networks and hosts
  - > L2 addressing is a first attempt at fine grain virtualization addressing

# Software Stack

- The Networking and FC stacks are separate entities
  - > both are in constant evolution
- The software stack is based on
  - > modular layering
  - > flexible policy
- Both the legacy stack and the virtualization layer provide transparency between local or remote peers
- The stack can take advantage of network interface assists but must also implement SW solutions for less sophisticated devices

# Length Field Impact

- The FCoE interface is an Ethernet interface, so it is essentially capable of dealing with protocols that do not specify the payload length in their headers
- The insertion and checking of FC CRC is a new interface centric function that depends on the payload length (implicitly or explicitly)
- For the purposes of locating the FC CRC (TX and RX) any of the three proposals would work for the data path

# Length Field Value

- A length field makes the system more robust by:
  - > Providing a length to the upper layer that is independent of any device driver defects or network interface faults
  - > Note that both CRCs will be implemented by the same data path on the same piece of silicon
  - > Similar weakness are well documented for TCP checksum offload, as it comes from the NIC, not from system memory
- Length field insertion for endnode generated frames is straightforward
- Length field saves cost in wide data path Rx implementation

# Timestamp Impact

- An end-to-end solution seems most appropriate for its higher value regardless of endpoint implementation cost
- FCoE may have renewed the concern but if there is a problem then it has to be addressed at the FC layer rather than through the encapsulation over Ethernet

# Addressing Model

- The host interface datapath needs to steer and isolate traffic, with FCoE being one of the variables
- Best practice is to demux layer N (or above) by using a layer N-1 header field, in this case L2
  - > Ethertype is the obvious choice
- The current virtualization precedent is to use different L2 addresses
- Network interfaces can parse and demux based on L2 address, Ethertype, or both
- Ethertype separation should be mandated regardless of L2 addresses to ensure that the stacks do not ever mix

# Addressing Model

- Both address uniqueness and scalability are being challenged by virtualization
  - > The FCoE OUI model is similar to current L2 virtualization practice
    - > Best suited to be accelerated in hardware
  - > The limitations of this practice can be fixed in the future by parsing higher layers of the IP stack, but FCoE as a non-routable protocol has no such option
    - > The FCoE OUI MAC could be used to provide such a level of indirection
  - > In summary, it seems wise to specify Ethertype demux plus building the dual address mechanism on top of that