Creating High Performance Clusters for Embedded Use
The Internet of Things has the capacity to create huge amounts of data
Gartner forecasts ‘35ZB of data from things by 2020’ etc

Putting the numbers to one side, we have seen a trend to:
› Provide more locally based filtering
› Execute preliminary data analysis, sometimes this is very complex
› Deliver an interim result

Examples include oil and gas surveying and Improvised Explosive Device (IED) image recognition
Desire is for supercomputer performance in the size of a shoebox
High Performance Computers (HPC)

- Trend for massively parallel distributed architectures
- Heterogeneous architectures: CPU and GPGPU
- Use high speed interconnects:
  - Proprietary 2D and 3D Torus
  - Infiniband or Ethernet
- Size, Weight and Power are not particularly important criteria. Neither is Cost!
High Performance Embedded Computers (HPEC)

For use in Embedded applications other factors dictate compromises:

- Size, Weight and Power (SWaP)
- Operating Temperature
- Granularity
- Ruggedness
- Open-ness

This is really why standards like VPX exist

Other considerations:

- More Real Time
- Deterministic
- Low Latency

Results in seconds and minutes rather than weeks or months
HPEC Implementations – Form Factor

**Before**
- VME
- 32 or 64 bit parallel
- Single Core
- Homogeneous

**Now**
- VPX, SFF, AMC
- Ethernet, PCI Express, RapidIO, Infiniband
- Multi core
- Heterogeneous

Lots more choice and complexity today
Serial Fabrics – not so simple

Typical VPX system showing multiple processing elements
Communication across appropriate fabrics
Software Applications

- The interconnect between servers is typically Ethernet based.
- Software packages from the HPC space tend to use TCP/IP socket APIs running on a Linux OS.
- A good example of this is Hadoop which is an open framework for distributed processing of large data sets across clusters of computers.

- The challenge has been how to utilize this ecosystem of applications in an embedded environment where PCI Express or RapidIO fabric interconnects might be used.
Our solution: FIN-S

- Emulates an Ethernet device over PCI Express or RapidIO
- From an application perspective, the interconnect is seen as an Ethernet network running over TCP/IP
- This shields the application from the underlying fabric and allows some useful side benefits:
  - Improved throughput with PCI Express and RapidIO (slide 11)
  - CPU utilization reduction (slide 12)
  - Best latency with RapidIO (slide 13)
Comparison Measurements

Comparison was done using:
- Processor board with a 10 Gigabit Ethernet adapter connected via a x8 Gen2 PCI Express link
- Processor board running FIN-S on a PCI Express Gen2 x4 fabric across the backplane
- Processor board running FIN-S on a RapidIO Gen2 (5 Gbps) x4 fabric across the backplane
Throughput vs Packet Size

Throughput Comparison

<table>
<thead>
<tr>
<th>Packet Size</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1k</th>
<th>2k</th>
<th>4k</th>
<th>8k</th>
<th>16k</th>
<th>32k</th>
<th>64k</th>
<th>128k</th>
<th>256k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth MB/s</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10G Ethernet</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIN-S PCIe Gen 2 x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIN-S RapidIO Gen 2 x4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CPU Utilization vs Packet Size

Utilization Comparison

CPU Utilization (%) vs Packet Size

- 10G Ethernet
- FIN-S PCIe Gen 2 x4
- FIN-S RapidIO Gen 2 x4
Latency Comparison

Latency (us)

Packet Size

10G Ethernet
FIN-S PCIe Gen 2 x4
FIN-S RapidIO Gen 2 x4
Example Implementations

- **4 x AdvancedMC Modules, AM 945/ x1x AdvancedMC modules based on Intel i7-3612QE**
- RapidIO fabric
- 1U DCCN proof of concept box

- **4 x 3U VPX Boards, TR 905/ x11 3U VPX boards based on Intel i7-3612QE**
- PCI Express fabric
- 6 slot 3U VPX Development System

Key is that we can demonstrate consistent results using the same application
Summary

- There is a real need to provide HPEC solutions
- Customer expectations are increasing:
  - Increased bandwidth
  - Reduced SWaP (and cost)
  - Lower Latency and Deterministic Performance
  - Easier to Scale
  - Leveraging relevant software from the commercial space
- FIN-S is one solution that can allow customers to base HPEC solutions on RapidIO and PCI Express fabrics without significant change

Thank you for listening