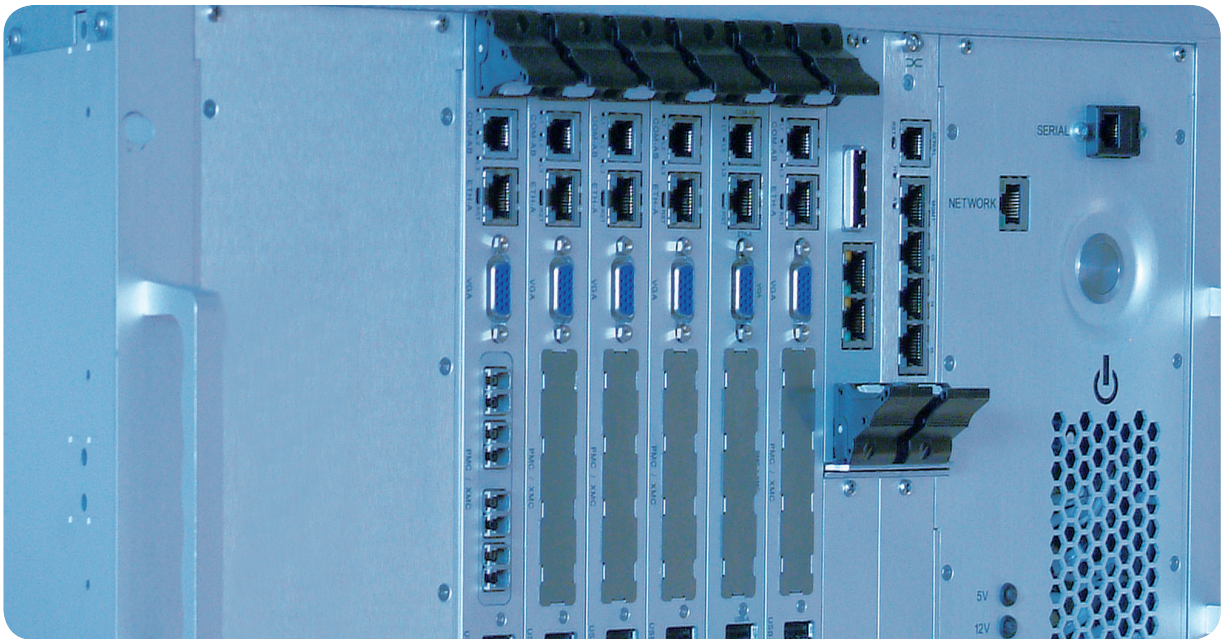# » Whitepaper «

Kontron VXFabric™

PCI Express Switch Fabric for High Performance Embedded Computing
(2nd edition: VXFabric™ meets gen3 PCIe)

Regis Odeye, Vincent Chuffart, Serge Tissot, Robert Negre

Kontron VXFabric™

**PCI Express Switch Fabric for High Performance Embedded Computing
(2nd edition: VXFabric™ meets gen3 PCIe)**

Simple and future-proof: Using TCP/IP over PCI Express as a data plane for High Performance Embedded Computing.
Current performance figures of TCP/IP on gen3 PCIe demonstrate data transmission bandwidth better than multiple 10GETH links. (see fig. 4, page 4 and fig. 12 page 12)

Abstract

This whitepaper introduces the Kontron VXFabric™ technology which implements TCP/IP protocol over the PCI Express infrastructure. It is currently deploying with Kontron VPX boards and embedded computers. Real life performance figures are demonstrated for this data plane solution proposed by Kontron. The technology insulates applications from the complex, low-level details of the current generation of PCIe silicon management.
With VXFabric™, the use of standard communication protocols, TCP/IP or UDP/IP, based on the socket API is the ultimate protection of customer application software investment. For years to come, by using Kontron VXFabric™, OEMs and customers will enjoy an optimized total cost of ownership (TCO) and have a direct migration path from their existing application deploying today. VXFabric™ is the cost effective bridge between current GETH on the backplane (VME, cPCI, VPX) into the next data plane generation: 10G and 40G Ethernet. Right now, VXFabric™ offers 10G and 40G performances in VPX compact computers featuring low power consumption and harsh environment capability. VXFabric™ addresses all fast and low latency peer-to peer inter computer node communication within a chassis.

TCP/IP on PCie: a wonderful opportunity

The advent of VPX opens a new era of rugged embedded computing. VPX allows board computers to move away from decades of parallel bus architectures on the backplane that seriously limited the CPU to I/O performance ratio. During the last 10 years, only Gigabit Ethernet did something to enhance the intra communication bandwidth on backplanes. In that period, board computing had to use several different hardware and software solutions for high speed serial link point-to-point connections between boards. Due to the lack of market traction, none of them is expected to survive the next decade.
Enters VPX, with an ecosystem defined for board sizes, connectors and signaling for the next 10 to 20 years. VPX connectors and backplanes can carry multi-gigahertz signals and allow to build computers where the bandwidth is no longer shared between boards.
High Performance Computing is the first domain to benefit from the tenfold increase in I/O bandwidth between computing boards offered by VPX backplanes. This enables a new breed of unparalleled applications for sensor data processing platforms used in radar, sonar, and general imaging. The I/O performance gap is such that only a fraction of the HPEC market actors really see the new generation of sensor computing application benefits in terms of accuracy, signal to noise improvement, cost and compactness of embedded computers.
Based on a solid standard (VITA46 and OpenVPX VITA64), OEMs and developers have to select the ideal communication protocols. With all the advantages of current standards such as PCI Express, Gigabit Ethernet, Serial Rapid IO and many others that can be used for intra-system communications, the challenge for OEMs is to choose an easy to use, yet fast and low latency communication protocol; this is not a simple task.
However, with VXFabric™, maybe the solution IS simple after all.
Amongst serial link technologies available today on VPX, Peripheral Component Interconnect Express (PCIe) is poised to bring demanding applications to a new level thanks to the whole IT market adoption of the technology. And with VXFabric™ bridging PCIe and TCP/IP together, application software can be protected from obsolescence for the next 20 years !

CONTENT:

1. PCI Express intrinsics

PCIe is a computer expansion card standard initially designed to replace the older PCI and PCI-X standards. The PCIe topology is based on point-to-point serial links, instead of a system-wide shared parallel bus architecture. PCIe technology has evolved recently from Gen2 to Gen3, doubling the effective throughput per lane from 4 Gbits/s to 8 Gbits/s. The effective raw bit rate has increased only from 5 GHz to 8 GHz, but by optimizing the efficiency of the raw encoding from 8b/10b to 128b/130b, the bandwidth is actually multiplied by two.

At the software level, PCIe preserves compatibility with PCI even if new capabilities have been added (Power Management, Advanced Error reporting, etc). In terms of bus protocol, PCIe communication is encapsulated in packets.

The link between two PCIe ports is composed of lanes used in aggregates from x1 to x32 in powers of two. The size of this aggregate is a compromise between the required bandwidth and the pin count. (important: lane and pin count drive the ultimate size of the final computer, an important factor in SWAP-C sensitive applications).

Lanes	Gen 1 (2.5Gb/s bit rate)	Gen 2 (5Gb/s bit rate)	Gen 3 (8Gb/s bit rate)
x1	250MB/s	500MB/s	1GB/s
x2	500MB/s	1GB/s	2GB/s
x4	1GB/s	2GB/s	4GB/s
x8	2GB/s	4GB/s	8GB/s

Figure 1: PCIe Maximum theoretical bandwidth

These features make PCIe an ideal solution not just to link high bandwidth I/Os to a processor unit, but also, become a native communication link between computing devices in a multiprocessor environment. PCIe switches implement all needed mechanisms for interconnecting CPU boards so as to allow transfers between memories (through DMA or PIO) of each board. Such mechanisms include PCI address translations and protection, DMA engines, Doorbell interruptions, scratchpad registers and flow control. All these features together implemented in PCIe switches are called Non-Transparent (NT) mode. This NT mode is required in any PCIe system architecture that implements peer to peer communication on PCIe.
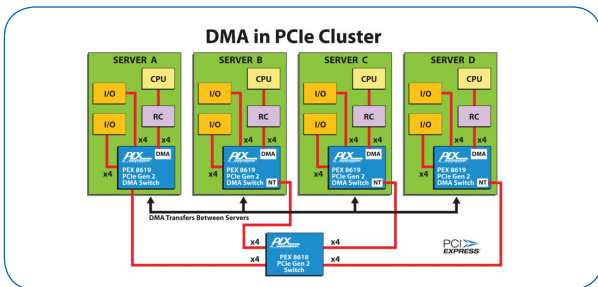

Figure 2: PCIe switched topology (Courtesy of PLX technology, Inc...)

2. Kontron VPX Hardware Building Blocks

2.1 Kontron VX30xx: 3U VPX SBC family

The Kontron SBC portfolio includes 3 upward compatible versions of 3U Core™ i7 single board computers: VX3030, VX3035 and VX3042/VX3044. Each of them implements a non transparent port on the PCIe and can be used in any payload slot of a VPX backplane with PCIe to exchange high speed data through VXFabric™.

VX3030
3U VPX, Intel® Core™ 1st Gen

VX3035
3U VPX, Intel® Core™ i7 2nd Gen

VX3044
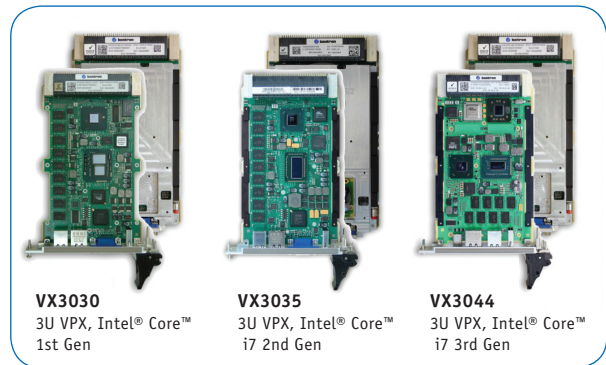3U VPX, Intel® Core™ i7 3rd Gen

Figure 3: Kontron 3U VPX SBC family

Each new generation offers a higher performance on PCIe using larger buses and higher speeds. However, each of these SBCs, can be used in a low end infrastructure (x2 PCIe or gen1 limited backplanes) and also be programmed to drive multiple x1 PCIe links to multiple payloads. In this case, VXFabric™ can still offer TCP/IP over PCIe and bring dedicated additional bandwidth to the application data plane, usually implemented only on GETH in small computers and shared with the control plane.

	VX3030	VX3035	VX3044
BP interface	x4 PCIe gen1	x4 PCIe gen2	x8 PCIe gen3
VXFabric™ BW on tcp/ip (iperf)	600MB/s	1300MB/s	>4000MB/s
Ethernet Equivalence	5*GETH	>1*10GETH or >8*GETH	=4*10GETH >34*GETH!

Figure 4: Kontron 3U VPX SBC family and data bandwidth performance

2.2 Kontron VX60xx: VPX twin CPU computing nodes

To implement HPEC cores in 6U VPX computers, Kontron develops 6U VPX boards featuring two CPUs. Their architecture is optimized to offer maximum bandwidth and computing power in this form factor.

Such a dual CPU 6U VPX board is managed as two distinct CPU nodes by VXFabric™, offering the user a uniform programming model and a perfectly scalable solution, whatever the form factor.

VX6060 and VX6080 are good examples of such computing node boards.

2.3 Kontron VX3905: PCIe and Ethernet Hybrid switch

The Kontron VX3905 is an essential part of the 3U VPX ecosystem. This 3U VPX PCIe and Ethernet Hybrid switch is compliant with the OpenVPX VITA65 profile SLT3-SWH-6F6U-14.4.1 and provides up to 24 Ports/32 Lanes PCIe Switch and a 9 Port Giga Ethernet Switch. VX3905 is available in air-cooled and conduction cooled builds.

It can be paired with Kontron 3U and 6U CPU boards to build the centralized topology based on the Gen1 or Gen2 PCI Express protocol. There is also a x4 PCI Express Cable downstream output on the front panel. VX3906 is a PCIe gen3 version of this switch and is expected to be released mid 2013.
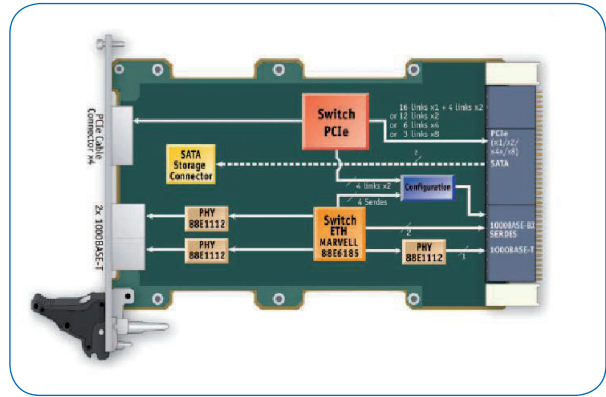

Figure 5: The Kontron VX3905 PCIe and Ethernet Hybrid switch

3. Computer fabric topologies

The Open VPX standard describes the elements to build two families of computer architecture:

» Centralized topologies rely on centralized backplanes with payload slots and switch slots profiles (same philosophy as in VITA31 for VME, or PICMG 2.16 in cPCI)
» Distributed topologies use simpler payload-only backplanes, with direct point to point connections between fixed links of several payload boards. There is no switch here.

The use of Distributed topologies fits very simple computers and is adequate when the computer architecture, control and data plane are perfectly defined and immutable. In this situation, significant deployments can enjoy cost saving by eliminating the use of a switch. This requires a custom backplane routing implementing the required point to point liaisons. The use of Centralized topologies offers the most performance and ease of use. It should be used in development phases where the computer data and control plane need to be reconfigured often to find the optimal workload and I/O distribution between boards. This is true for small and large computers alike. Centralized topologies is almost compulsory for large HPEC architecture with a need for health management, redundancy and load balancing.

3.1 Distributed Topologies

For simple configurations, the use of a switch can be avoided by using direct PCIe links from the first CPU slot to the second (and third) slots. The figure below describes the standard OpenVPX topologies for the 2 slots and 3 slots distributed topologies.
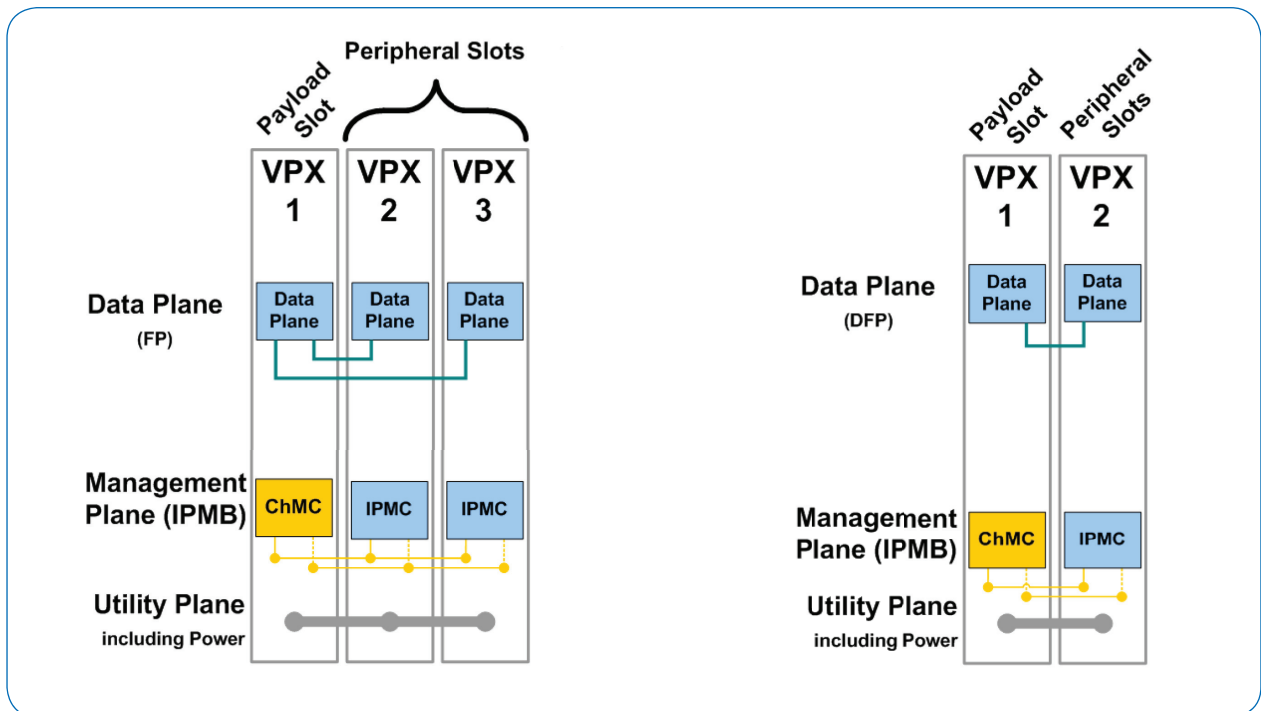

Figure 6: Example of distributed topologies, 3 slot and 2 slot x8 distributed

3.2 Centralized Topology (6 payloads on PCIe x4)

The centralized topology configuration utilizes a Kontron VX3905 PCIe and Ethernet Hybrid switch with 6 x4 PCIe lanes to interconnect up to six Kontron SBCs (slots 1 to 6). The PCIe switch occupies VPX slot 7. Any CPU can communicate with any other CPU. The VXFabric™ traffic is balanced through the Kontron VX3905 PCIe ports. Compared to the previous distributed architecture, the PCIe port of the system controller in slot 1 is off-loaded of all the traffic between the other boards.

Note: Slot 8 is populated with a VX3910, a GETH switch which implements the control plane. Each board can communicate with each other board. Programming the switch allow true dual star redundancy or various virtual networks profiles.
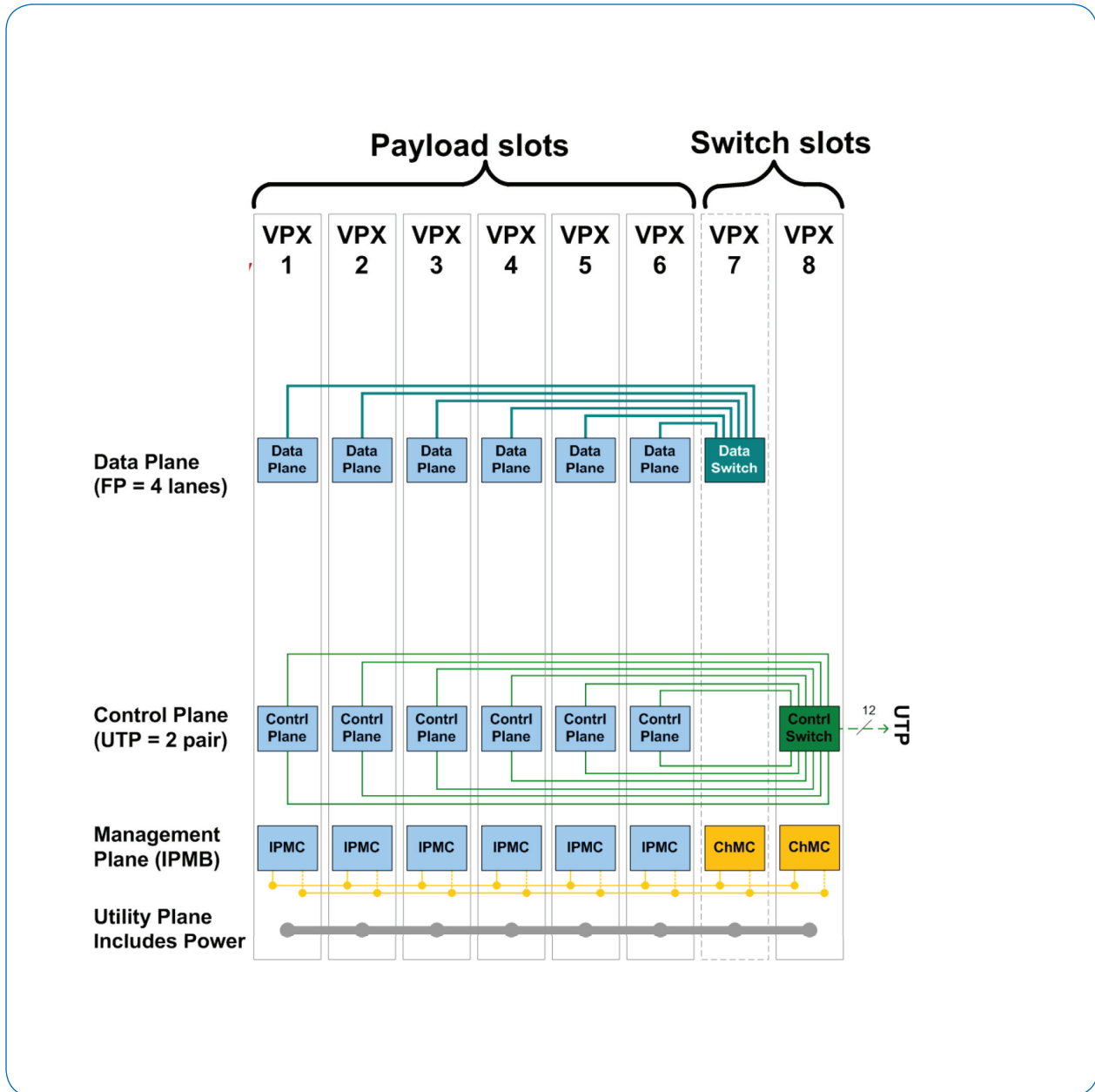

Figure 7: Single Star Centralized Topology: 6 * PCIe x4

3.3 Centralized Topology (12 CPUs on PCIe x2)

This configuration requires a VX3905. This PCIe Switch is able to service a single star topology of 12 PCIe x2 lanes connecting six Kontron VX60xxs while sticking to a 3U VPX backplane to implement the computer. This approach is cost effective and offers integrators the possibility to implement multiple customized computers I/O with minimal risk.

In this configuration each CPU on the Kontron VX60xxs (CPUA and CPUB) are connected to VXFabric™. Any CPUA or CPUB can communicate directly with any other CPUA or CPUB: it is the centralized equivalent to a full mesh topology.
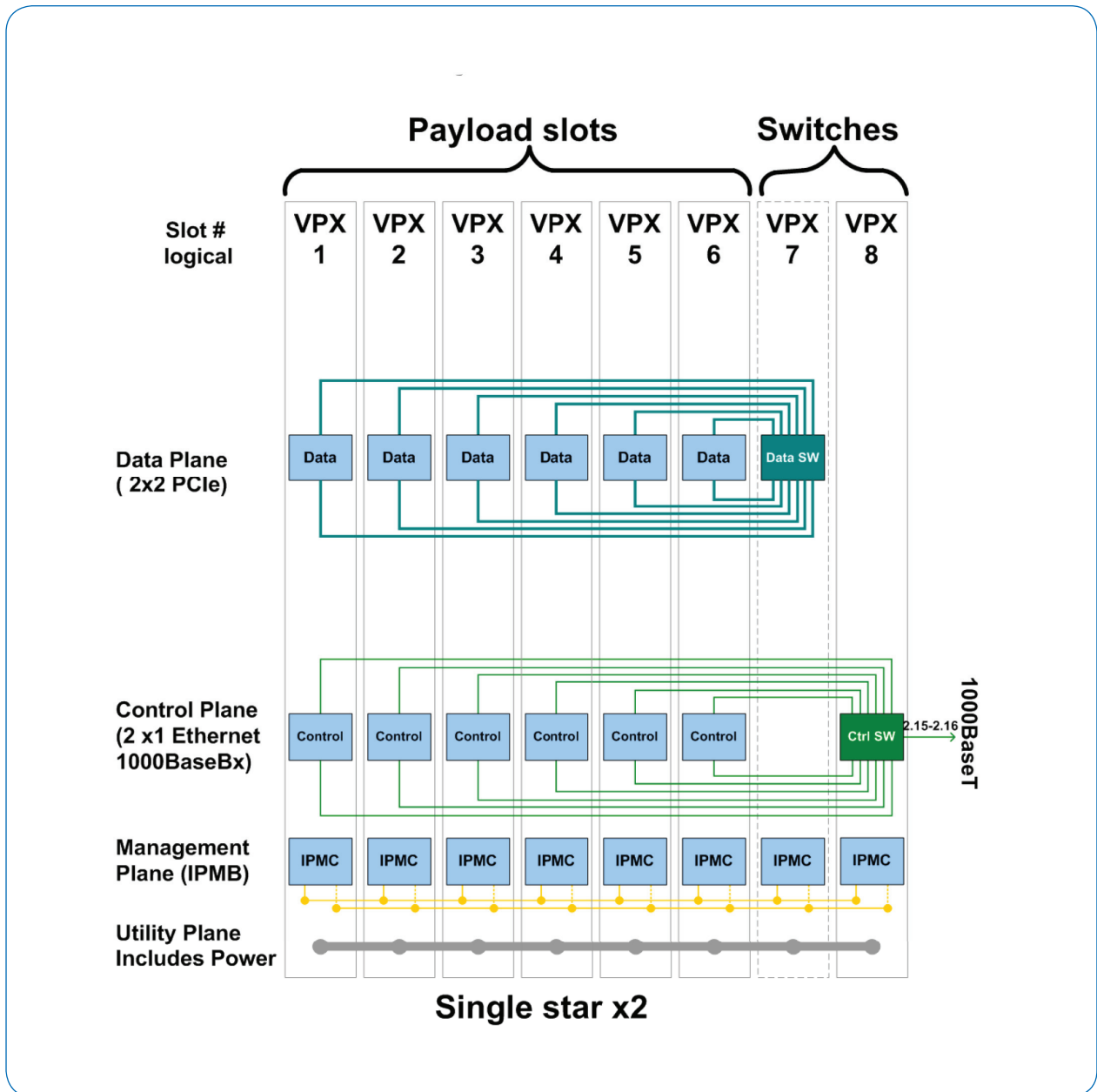
Figure 8: Single Star Centralized Topology

4. VXFabric™ Software Architecture and API

Kontron VXFabric™ is an open infrastructure which implements efficient inter-board communication at hardware speed. The architecture is compliant with the OpenVPX standard (VITA65) which defines two main hardware topologies of the backplane: distributed and centralized topologies.

In its current form, VXFabric™ can simultaneously interconnect up to 12 nodes via PCI Express. The physical interconnect through the backplane can be made in various implementations according to the choice of backplane style:

» A distributed PCIe backplane can be used without the need of any PCIe switch. (see Backplanes examples). In this case, very compact architectures can be proposed, as long as the relevant data path between nodes is implemented in the backplane. This data path is mostly application dependent, making distributed backplane the right solution for cost effective deployments.
» When a higher bandwidth and more flexibility in the data plane is required, a centralized topology is recommended. With a PCIe switch, such as the Kontron 3U VPX PCIe and Ethernet Hybrid switch VX3905, it is possible to interconnect up to 12 nodes with the VXFabric™ via PCI Express and establish communications links between any nodes using the same backplane. This approach fits well the HPEC application domain, as well as lab use where multiple application data path can be evaluated with the same equipment.

The following sections describe the multiprocessor architecture implementing Kontron VXFabric™. From the hardware point of view, the architecture is based on several CPU boards, each featuring several processing cores, interconnected through PCIe via the VPX backplane, using a PCIe switch.

Software wise, Kontron VXFabric™ is equivalent to an Ethernet network infrastructure mapped over a switched PCIe express fabric. It implements the layers allowing the user to handle the communication with an IP socket programmatic interface. This API allows direct access to all classic protocols like TCP or UDP. Furthermore, Kontron VXFabric™ requires no modification of existing applications, which helps reduce development efforts and simplifies migration to the new VPX architecture.

4.1 Kontron VXFabric™ and the IP Protocol

The standard user programming model of Kontron VXFabric™ is based on the IP protocol and implements a socket layer API through an emulation of an Ethernet interface over PCIe (similar to implementations of pseudo-Ethernet found in virtual machines).

This API is the main reason why the compatibility of existing applications with Kontron VXFabric™ is guaranteed. The task to migrate from a Gbit Ethernet TCP/IP infrastructure towards Kontron VXFabric™ avoids the low level complex and proprietary APIs (like most of serial RapidIO™ or Infiniband™ implementations for example) and is really straightforward. VXFabric™ is available under Linux for all Kontron VPX CPU boards and it is designed to be portable on other operating systems offering a modern TCP/IP stack as well as other architectures (e.g: FPGA). This implementation is scalable (it did not required much effort to recently evolve towards PCIe gen3 silicon). Furthermore, VXFabric™ does not require any other infrastructure than the VPX backplane and a VPX PCIe switch to interconnect VPX boards. 100% of the necessary hardware and silicon involved comes from the main IT market and not at the mercy of a little group of suppliers.
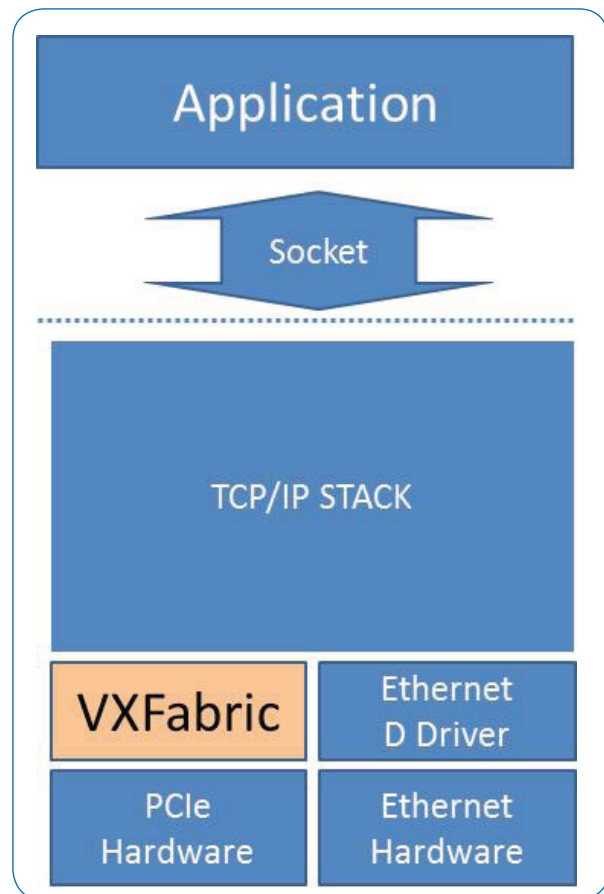

Figure 9: VX Fabric™ implementation

4.2 VXFabric™ Setup

The following paragraphs give a step-by-step description of the VXFabric™ initialization and usage. In a typical computer, these steps would be part of the standard boot sequence and become invisible to the casual user.

To setup the Kontron VXFabric™ on a Kontron VPX CPU board, only a few simple steps are required. The procedure is the same for centralized or distributed topologies.

Once the first CPU in the system, located in slot 1, has booted the Linux operating system, the Ethernet emulation over VXFabric™ has to be initialized.

The first step is to get some low level information about the VXFabric™ through the 'vxfabric' command line interface:

After the first board has booted the VXFabric™, all the boards reach the READY status.

```
ID_H  SLOT  STATUS  CPU  SYSC    MEM_AREA( SIZE )       IO_AREA   ( SIZE )
----------------------------------------------------------------------------
 1  *   1    READY    A    *   0x08000000(0x4000000)      NONE   ( NONE )
 7      4    READY    A        0xc0000000(0x4000000)  0xf7800000(0x20000)
 9      5    READY    A        0xc8000000(0x4000000)  0xf7a00000(0x20000)
11      6    READY    A        0xd0000000(0x4000000)  0xf7c00000(0x20000)
----------------------------------------------------------------------------
```

"modprobe vxeth" is then used to load the vxeth kernel module.

This command generates a new pseudo-Ethernet interface vxeth0, which is similar to a standard eth# Ethernet interface. The user may then check the interface set-up and initialization:

```
client9# ifconfig vxeth0
vxeth0    Link encap:Ethernet  HWaddr 02:00:00:00:00:01
          BROADCAST MTU:65500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Such a pseudo-Ethernet interface is managed like any other Ethernet interface. The next step is to allocate a local sub-network address. This is done with the *ifconfig* command:

ifconfig vxeth0 192.168.20.1

In Linux Fedora distributions or derivatives, this Ethernet interface configuration is managed by the "network" service. It does the setup of the interfaces according to configuration files (here */etc/sysconfig/network-scripts/ifcfg-vxeth0*.)

It is possible to check the status of the interfaces using the "*netstat*" command:

```
client5# netstat -i
Kernel Interface table
Iface       MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
bond0      1500   0   382872     0      0      0   391990     0      0      0 BMmRU
eth0       1500   0   204225     0      0      0   206587     0      0      0 BMsRU
eth1       1500   0   178647     0      0      0   185403     0      0      0 BMsRU
eth2       1500   0        0     0      0      0        0     0      0      0 BMU
vxeth0    65500   0  5372700     0      0      0 18477441     0      0      0 BMRU
lo        16436   0       36     0      0      0       36     0      0      0 LRU
```

Note:
» The MTU is set by default to 65500 bytes to reach the maximum bandwidth.

On the other nodes, the same steps have to be performed, with different IP addresses on the same local sub-network address.

The IP addresses of all the nodes can now be added in the /etc/hosts file.

```
client5# more /etc/hosts
# hostname vx6060 added to /etc/hosts by anaconda
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
192.168.1.1    server
192.168.20.1    node1
192.168.20.2    node2
192.168.20.3    node3
192.168.20.4    node4
192.168.20.5    node5
192.168.20.6    node6
### netconfig updates ####
192.168.1.12    client12
192.168.1.11    client11
192.168.1.10    client10
192.168.1.7    client7
192.168.1.6    client6
192.168.1.5    client5
192.168.1.4    client4
192.168.1.3    client3
192.168.1.2    client2
192.168.1.9    client9
192.168.1.8    client8
### netconfig ends ####
```

The interconnection is tested through the "*ping*" command:

```
client5# ping node2
PING node2 (192.168.20.2) 56(84) bytes of data.
64 bytes from node2 (192.168.20.2): icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from node2 (192.168.20.2): icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from node2 (192.168.20.2): icmp_seq=3 ttl=64 time=0.020 ms
64 bytes from node2 (192.168.20.2): icmp_seq=4 ttl=64 time=0.019 ms
64 bytes from node2 (192.168.20.2): icmp_seq=5 ttl=64 time=0.022 ms
```

When every node of the VPX system is configured correctly and initialized with a valid internet address via Kontron VXFabric™, the "vxfabric" command output becomes:

```
client5# vxfabric -a
-----------------------------------------------------------------------------
  VX FABRIC 0 configuration
-----------------------------------------------------------------------------
  ID  H  SLOT  STATUS   CPU  SYSC   MEM_AREA( SIZE )         IO_AREA   ( SIZE )
-----------------------------------------------------------------------------
   1     1    CONNECT    A    *    0xc0000000(0x4000000)       NONE   ( NONE )
   7  *  4    CONNECT    A         0x10000000(0x4000000)    0xf7c00000(0x20000)
   9     5    CONNECT    A         0x88000000(0x4000000)    0xc5a00000(0x20000)
  11     6    CONNECT    A         0x90000000(0x4000000)    0xc5c00000(0x20000)
-----------------------------------------------------------------------------
```

4.3 Standard Network services

Since VXFabric™ code lies below the TCP/IP stack, the standard network services are supported "de facto" thanks to the Ethernet emulation. After the vxeth setup has been executed, the user can configure any TCP or UDP network service such as: ftp, telnet, ssh, NFS, scp, etc, according to their needs.

```
client5# ssh node3
Warning: Permanently added the RSA host key for IP address '192.168.20.3' to the list of known hosts.
Last login: Thu Nov 25 11:04:51 2010 from server
client7#
client7# scp /boot/vmlinuz-2.6.32.14-10176.vx6060.fc12.i686.PAE node2:/tmp
Warning: Permanently added the RSA host key for IP address '192.168.20.1' to the list of known hosts.
vmlinuz-2.6.32.14-10176.vx6060.fc12.i686.PAE              100% 3565KB   3.5MB/s   00:00
client7#
logout
client5# ll /tmp
total 3568
-rwxr-xr-x 1 root root 3650176 2010-11-25 16:04 vmlinuz-2.6.32.14-10176.vx6060.fc12.i686.PAE
```

Example: using ssh and scp on top of VXFabric™ TCP/IP connection.

Also because of the code architecture, the standard *iperf* tool is used unmodified to evaluate and measure the performances, as described later.

4.4 DATA communication through Socket API

There is no difference when using the socket API on top of Kontron VXFabric™ or on top of a standard Ethernet interface. The datagram socket (UDP) or streaming socket (TCP) API is the preferred way to address the Kontron VXFabric™ which is usually dedicated to DATA communication.

A good coding example can be found in the source code of the "iperf" (open source GPL) tool.

Note: This tool has been used below to get benchmark and throughput measurement of the socket implementation found in this white paper.

4.5 Other API alternatives

For less portable applications and huge data sets, RDMA is also used in very big computers, scaling well beyond dozens of nodes (such as Supercomputers or computing farms). In this case, OFED (Open Fabric) is proposed as a communication stack and API. OFED's goal is to offer all the possible data transfer APIs on all kinds of hardware.

However, this has a price, and the complexity of the software stack speaks for itself.
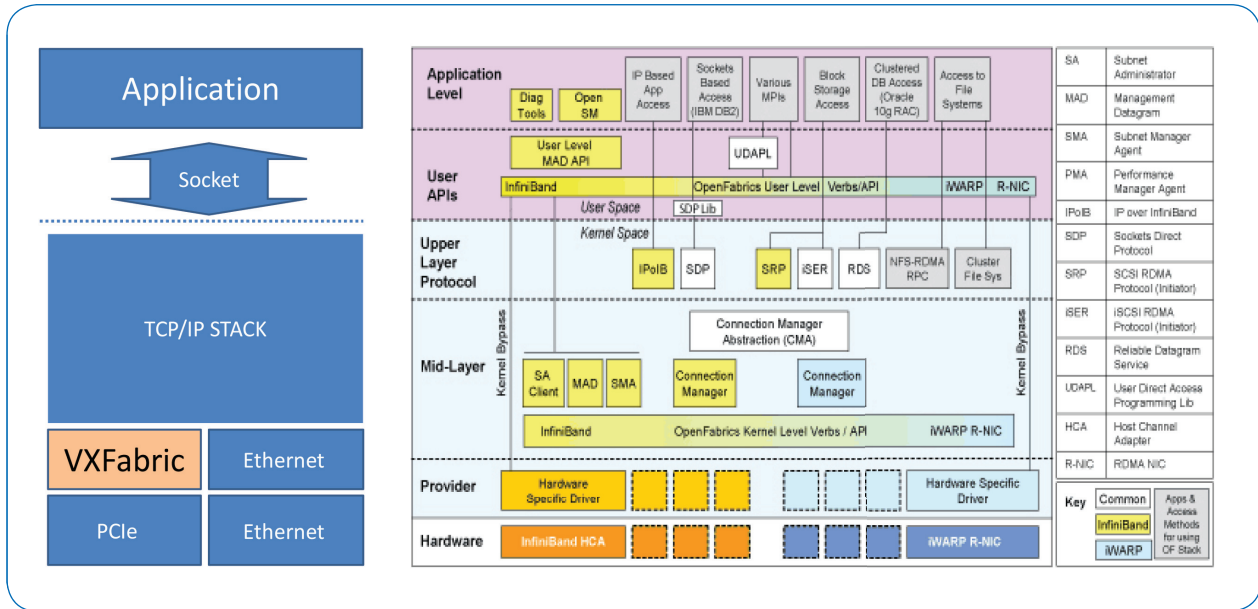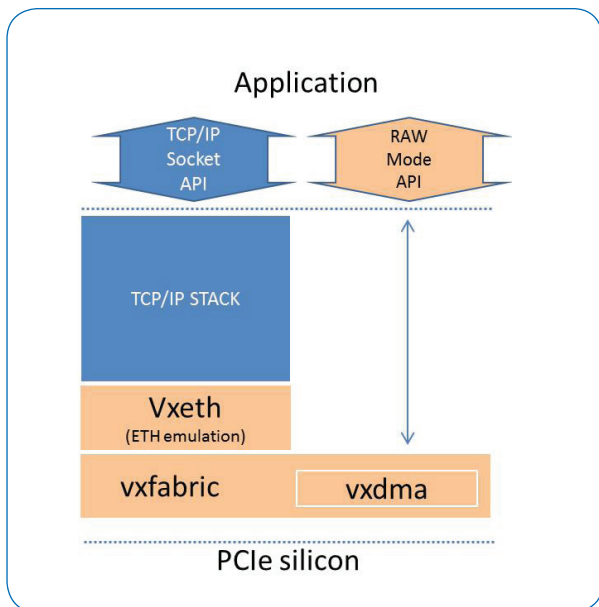

Figure 10: VXFabric™ and OFED software stacks

Figure 11: VXFabric™ TCP and RAM API

4.6 VXFabric™ Raw Mode

For specific communication links where real-time data transfers have to be used, an application can choose to use the VXFabric™ raw mode to move large data sets directly from user memory to user memory. Driving the PCIe silicon DMA engines, the utility layers of VXFabric™ take care of handling the low level programming and the management of scatter/gather DMA job lists while the VXFabric™ infrastructure maintains the necessary system-wide address mapping coherency.

With this raw mode, the user cannot rely on all the intrinsic features of TCP/IP communication (guaranteed delivery, data check, flow control) and must find other means to synchronize between the multiple parts of a distributed application.

5. VXFabric™ Performance Figures

Sustained throughputs have been measured with the standard iperf tool.

PCIe gen3 performances have been demonstrated on VX3042/VX3044 cards hosting the Intel® Core™ i7 3rd-Gen processor (Ivy Bridge). For PCIe gen2, results were obtained for communications between VX3035 cards based on the Intel® Core™ i7 2nd-Gen processor (Sandy Bridge). And the VX3030/VX6060 cards, featuring the Intel® Core™ i7 1st-Gen (Arrandale CPU), were used to exercise the bandwidth in PCIe gen1 mode. During the measurements, Turbo boost and Hyperthreading were disabled.

	Sustained TCP bandwidth	CPU load TCP (load as a percentage of one multicore CPU)	Sustained raw bandwidth (1)
PCIe x8 gen3	4.4 GBytes/s	10% (quad core)	5.6 GBytes/s
PCIe x4 gen3	2.5 GBytes/s	6% (quad core)	2.8 GBytes/s
PCIe x4 gen2	1.3 GBytes/s	8% (dual core)	1.5 GBytes/s
PCIe x4 gen1 or x2 gen2	0.6 GBytes/s	5%	0.8 Gbytes/s

(1): Very light CPU load, transfer handled directly by switch DMA engine.

Figure 12: VXFabric™ sustained bandwidth and CPU usage

6. Conclusion

VXFabric™ is a technology deployed in many Mil & Aero applications both for high performance embedded computing (HPEC) with 12 quad-processor nodes, or in SWAP-C configurations with a few computer nodes and many I/Os. The PCIe pervasion in all computer technologies and the TCP/IP standard, used everywhere, are positioning VXFabric™ as the most efficient, inexpensive and perennial technology for switch fabric in embedded systems.

About Kontron

Kontron is a global leader in embedded computing technology. With more than 40% of its employees in research and development, Kontron creates many of the standards that drive the world's embedded computing platforms. Kontron's product longevity, local engineering and support, and value-added services, helps create a sustainable and viable embedded solution for OEMs and system integrators.

Kontron works closely with its customers on their embedded application-ready platforms and custom solutions, enabling them to focus on their core competencies. The result is an accelerated time-to-market, reduced total-cost-of-ownership and an improved overall application with leading-edge, highly-reliable embedded technology.

Kontron is listed on the German TecDAX stock exchanges under the symbol "KBC". For more information, please visit: **www.kontron.com**

CORPORATE OFFICES

Europe, Middle East & Africa
Oskar-von-Miller-Str. 1
85386 Eching / Munich
Germany
Tel.:+49 (0) 8165 / 77 777
Fax: +49 (0) 8165 / 77 219
info@kontron.com

North America
14118 Stowe Drive
Poway, CA 92064-7147
USA
Tel.:+1 888 294 4558
Fax: +1 858 677 0898
info@us.kontron.com

Asia Pacific
17 Building,Block #1, ABP.
188 Southern West 4th Ring Road
Beijing 100070, P.R.China
Tel.:+86 10 63751188
Fax: +86 10 83682438
info@kontron.cn